ISSN 0242-9799

LES CASSETTES

TRACE

TIREZ PLUS DE VOTRE TRS-80* ET DE VOTRE POQUETTE SHARP

NUMERO 7

Essai matériel : kit double densité, p 19 • Des monstres mathématiques, p 20 • Parler-vous franssais, p 24 • Des rubans sans fin, p 28



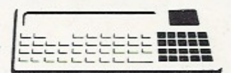
• La trace de vos variables, p 29 • Un



mini-zap, p 32 • Réflexions, p 35 • Vitesse V, p 38 • Améliorez la vitesse, p 43 • L'amorce, p 45 • Sésame ouvre-toi, p 46 • Des extensions à votre Basic, p 48 • Un



tri plus efficace, p 51 • Les lignes vitales, p 55 • RUBRIQUE POQUETTES : Désassemblez, p 57 • Essai PC 1245 et CE 153, p 59 • Zéro de fonctions,



p 60 • Bogues, p 60 • Jumpman, p 61 • La ville dans la poche, p 62 • Bi-mots, p 64 ■

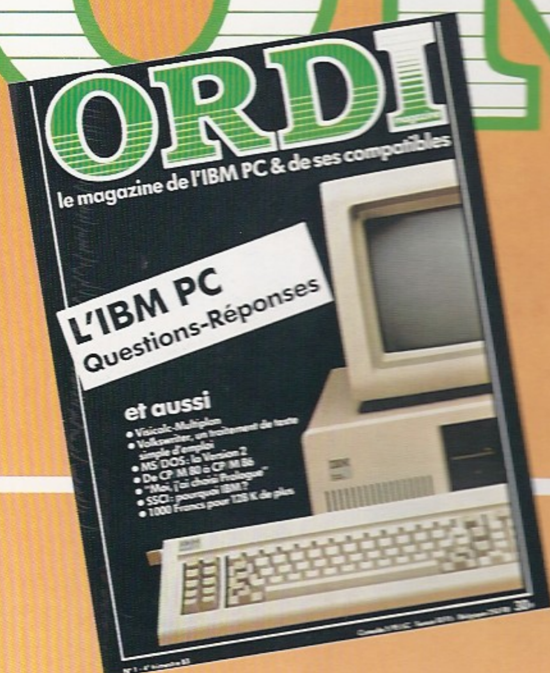
N° 7 - 3e trimestre 1983

Canada : 6,95 $ - Belgique : 230 FB - Suisse : 12 FS

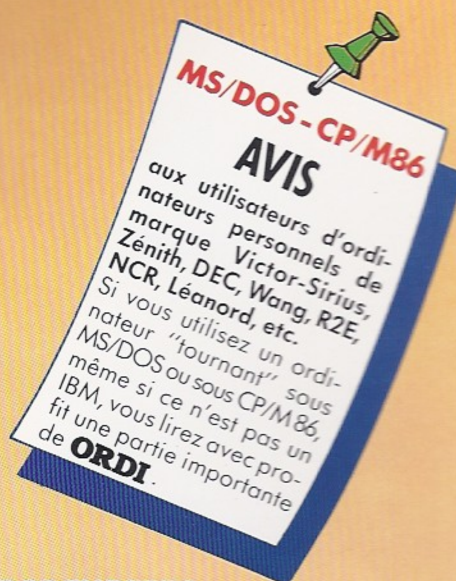
30 F

ORDI magazine

le magazine de l'IBM PC



Si vous utilisez un IBM PC ou si vous comptez en acheter un, sachez que **ORDI** a été créé pour vous. **ORDI** vous permet de mieux connaître votre ordinateur personnel et d'en tirer le maximum. **ORDI** vous tient au courant de toutes les nouveautés et vous fournit des programmes, des astuces et de nombreuses idées d'utilisation.



ORDI VOUS EVITE DES ACHATS DECEVANTS
Le prestige d'IBM est tel que dès à présent une multitude d'entreprises proposent des logiciels, des cartes d'extension et des périphériques pour le PC ; **ORDI** vous aide à détecter parmi tous ces produits ceux qui sont bien adaptés à vos besoins.

ORDINATEUR ET TERMINAL
ORDI étudie l'IBM PC sous toutes ses coutures : en tant qu'ordinateur autonome bien sûr mais aussi en tant que maillon d'un réseau local ou terminal d'une grosse unité centrale.

DES EXPERTS
ORDI est rédigé par des experts choisis parmi ceux qui connaissent le mieux l'ordinateur personnel d'IBM. Ne vous privez pas de leur expérience et de leurs conseils.

ABONNEZ-VOUS
... et profitez de notre tarif de lancement.

ORDI, le complément indispensable de votre IBM PC

BON DE COMMANDE
à retourner à ORDI MAGAZINE, 8 rue Saint-Marc, 75002 PARIS

Nom ______________________ Profession ______________________
Adresse ____
Pays ______________ Code postal __________ Ville ______________

☐ Je désire m'abonner à ORDI MAGAZINE 4 n^{os}, à partir du n° 1 ☐ (Tarif France 100 FF ; Etranger* 115 FF ; par avion 185 FF)
(actuellement ORDI MAGAZINE est trimestriel)

☐ Je désire recevoir le n° 1 ☐ de ORDI MAGAZINE. (prix d'un n° 30 FF ; Etranger* 35 FF ; par avion 50 FF)

Ci-joint mon règlement indispensable par chèque bancaire ☐ chèque postal ☐ virement* ☐

* Pour les pays autres que la France, utiliser un virement en FF compte Crédit Lyonnais Paris n° 30002 00402 8505 W.

TRACE

N° 7

Sommaire

Editorial	3
Courrier des lecteurs	10
Magazine	13
Côté court	15
STEP -1	16
Trucs à brac	17
Essai matériel : Kit double densité	19
De vrais petits monstres	20
Parler-vous un bon franssais ?	24
Des rubans sans fin	28
Gardez trace de l'usage de vos variables (II)	29
Mini-Zap deviendra grand	32
La réflexion mise en boîte	35
Des cassettes avec un grand V	38
Améliorez la vitesse de vos enregistrements sur cassette	43
Sésame : ouvre-toi !	46
Ouvrez votre modèle 1 sur le monde extérieur (III)	47
Cocktail d'extension Basic	48
Un tri encore plus efficace	51
Lignes vitales	55
Et désassemblez maintenant !	57
Un nouveau poquette : le Sharp PC-1245	59
Un autre clavier pour votre poquette	59
Lorsqu'une fonction s'annule ...	60
Bogues, trucs, erreurs (2e cuvée)	60
Jumpman, le saute-fantômes	61
La ville dans la poche	62
Un mot peut en cacher un autre	64
Le Dos de A à Z	65
Index	66

Editorial

Lorsqu'à l'été 1977 Tandy annonça son TRS80 modèle 1, le succès en fut immédiat. Non que la concurrence manquât : Commodore avait annoncé son PET, et Apple son Apple 2. Mais le premier ne devint réellement disponible qu'à l'automne, et le second était loin de disposer du gigantesque réseau de commercialisation des boutiques Radio-Shack. Ceci, joint à un prix fort raisonnable, eut pour effet la vaste diffusion du modèle 1 et des accessoires, tant logiciels que matériels, qui lui étaient destinés.

Hélas ! lorsque Texas Instruments voulut pénétrer sur le marché que se partageaient Tandy, Apple et Commodore (par ordre décroissant des parts du marché américain), ces derniers décidèrent de lui opposer quelques obstacles, par exemple en laissant entendre que le TI 99/4 de leur concurrent émettait un peu trop d'interférences radio. Message reçu, la FCC (Federal Communication Commission) bloquait pour quelque temps l'intrus, et participait ainsi à son échec commercial.

La riposte ne se fit pas attendre : le TRS80 modèle 1, l'Apple 2 et le Commodore PET se retrouvèrent à leur tour sur la sellette de la FCC. Les deux derniers ne se tirèrent pas trop mal des problèmes administratifs de la FCC, mais le TRS80 en souffrit : Tandy dut introduire le modèle 3, qui en plus de la conformité incontestable aux règlements de la FCC, présentait de nombreux avantages techniques par rapport à son prédécesseur le modèle 1.

Au détriment toutefois du prix : Tandy, qui avait pendant plus d'un an vendu l'ordinateur complet le moins cher, vendait maintenant un ordinateur qui n'était plus le meilleur marché, surtout face à Sinclair et Timex.

Simultanément, Tandy se réorientait vers les utilisations professionnelles : le modèle 2 était destiné à être l'arme absolue sur ce terrain, le modele 3 protégeant les flancs de son offensive. C'est à cette époque que Tandy, jusque-là leader incontestable et incontesté du marché, se retrouvait second derrière Apple, malgré l'accueil très réservé reçu par l'Apple 3. Aujourd'hui, IBM semble départager les deux rivaux, du moins en ce qui concerne les ventes de l'année 1983 sur le secteur professionnel ; Commodore est leader sur le secteur domestique, juste devant Tandy et son TRS couleur, l'impact non négligeable du Sinclair pouvant être considéré dans une autre catégorie. Il faut d'ailleurs reconnaître qu'autant Commodore que Tandy furent surpris en 1982 du succès aux Etats-Unis de leurs machines respectives (VIC et TRS Couleur), alors que le marché le plus important leur semblait être les utilisations professionnelles.

Avec ses nouvelles annonces de 1983, Tandy se retrouve bien armé pour faire face sur plusieurs fronts : côté domestique, le Couleur et le dernier-né MC-10, alias Matra Alice sont bien placés pour se défendre face aux Commodore-64, TI 99/4A et autres Sinclair ou Atari, du moins si Tandy arrive à les fabriquer et à les vendre à un prix assez bas ; côté professionnel, le modèle 4 dispose de tous les atouts du défunt modele 3, mais aussi du standard CP/M et donc d'une ouverture plus large vers les fournisseurs indépendants, en attendant qu'une version "portable" vienne compléter prochainement la version "de table" ; toujours du côté professionnel, la situation des modèles 12 et 16B est sans doute moins claire, du moins tant qu'il n'existe pas une base suffisante de logiciels pour les ordinateurs disposant comme le 16B du processeur 68000 16/32 bits ; enfin, dans un créneau légèrement différent, l'impact du modèle 100 semble considérable.

TRACE

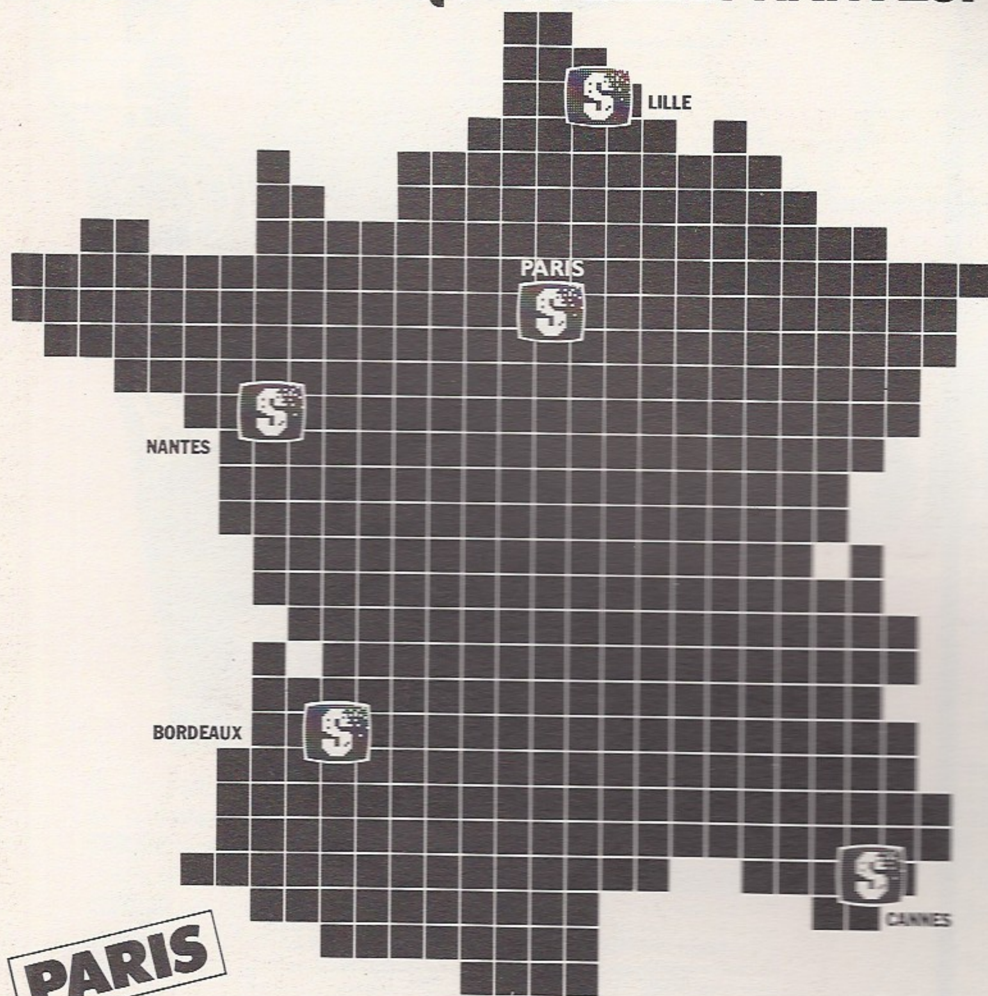
Rédacteur en chef : Alain Pinaud - *Editeur :* Bernard Savonet - *Directeur de la publication :* Jean-Pierre Nizard - *Maquette :* Studio Agraph - *Secrétariat :* Nicole Aleman.
Rédaction et abonnements : Editrace, 8, rue Saint-Marc, 75002 Paris - *Régie publicitaire :* Force 7, Anne Jourdan, 39, rue de la Grange-aux-Belles, 75484 Paris Cedex 10 - Tél. : 238.66.10 - *Diffusion auprès des boutiques informatiques et des librairies :* Edition du PSI, BP 86 - 77402 Lagny-sur-Marne Cedex.

Abonnements 4 numéros (un an) : France 105 FF ; Etranger 120 FF ; par avion 170 FF.

Ont collaboré à ce numéro : Kaarina Alain, Etienne Angel, André Beny, Josué Bonifas, Ivan Boussion, Roger Brousmiche, Roger Buvat, J.-F. Chapin, Paul Courbis, Jacques Dannenmuller, E. Ferrier, Michel Fourcade, Anne Fournier, Benoît Gilon, Daniel Glazman, Gérald Grandpierre, Roger Henrotay, Louis Martin, Serge Pinaud, Roger Sawicki, Christophe Schlick, Nicolas Spinga.

SIVEA® SIVEA® SIVEA®

PARIS-2 BOUTIQUES. LILLE. NANTES. BORDEAUX. CANNES.



LILLE
21 bis, rue de Valmy 59000 LILLE.
Tél. : 20/ 57.88.43 -
TÉLEX : 110 146
Ouvert du Mardi au Samedi sans interruption de 9 H 30 à 18 H 30.
Ouvert le Lundi de 13 H 30 à 18 H 30.
Parking assuré Place de la République (entrée par le Bd de la Liberté). Métro : République.

NANTES
21 A, Bd G. Guist'hau - BP 388.
44013 NANTES CEDEX.
Tél. : 40/ 47.53.09
Ouvert du Mardi au Samedi sans interruption de 9 H 30 à 18 H 30.
Ouvert le Lundi de 13 H 30 à 18 H 30.

BORDEAUX
Immeuble de la Croix du Palais.
Rue du Corps Franc Pommiès.
Meriadeck.
33081 BORDEAUX CEDEX.
(face à la nouvelle préfecture régionale).
Tél. : 56/ 96.28.11 - TÉLEX : 560.376
Parking assuré sous le centre commercial.
Ouvert du Mardi au Samedi sans interruption de 9 H 30 à 18 H 30.
Ouvert le Lundi de 13 H 30 à 18 H 30.

CANNES
14, Bd de la République.
06400 CANNES.
Tél. : 93/ 39.29.09 -
TÉLEX : 461 760.
Parking assuré Place de la Gare.
Ouvert du Mardi au Samedi de 9 H à 12 H 30 et de 14 H 30 à 19 H.
Ouvert le Lundi de 14 H 30 à 19 H.

De nombreux produits pour TRS-80 et Vidéo-Génie (Matériels, logiciels, librairie, revues) sont présentés en permanence dans les boutiques **SIVEA Informatique** : n'hésitez pas à consulter votre boutique SIVEA. Certains produits, disponibles dans notre entrepôt central, peuvent être obtenus sur demande dans votre boutique : demandez la liste de ces produits.

PARIS

Boutique Ordinateurs pour l'entreprise
31, bd des Batignolles.
75008 PARIS.
Tél. 522.70.66 - TÉLEX : 280 902.
Ouvert du Lundi au Samedi sans interruption de 9 H 30 à 18 H 30.

Boutique Ordinateurs domestiques.
33, bd des Batignolles.
75008 PARIS.
Tél. : 522.70.66 - TÉLEX : 280 902
Ouvert du Mardi au Samedi sans interruption de 9 H 30 à 18 H 30.

Service après-vente
33, rue de Moscou. 75008 PARIS.
Tél. : 293.02.22 - TÉLEX : 280 902
Ouvert du Lundi au Vendredi sans interruption de 9 H 30 à 18 H 30.
(Parking assuré au 43 bis, Bd des Batignolles. Métro : Rome-Place de Clichy).

NOUVEAU

LE CATALOGUE SIVEA VIENT DE PARAITRE !

Ce nouveau catalogue est entièrement consacré à l'**informatique domestique** : les matériels, les périphériques, les logiciels (jeux, utilitaires, langages, gestion familiale,...), les livres, les revues, etc... 80 pages (Format 21 x 29) pour découvrir les nouveautés et les grands classiques de l'informatique domestique !

Un second catalogue SIVEA entièrement consacré à l'informatique pour l'entreprise et les professions libérales paraîtra avant la fin de l'année.

BON DE COMMANDE

TR 10

Bon de Commande pour recevoir un Catalogue SIVEA INFORMATIQUE DOMESTIQUE à retourner à SIVEA S.A. 31, Bd des Batignolles, 75008 PARIS, accompagné d'un règlement (chèque uniquement) de 25 F.

NOM ______________________

PRÉNOM ______________________

ADRESSE ______________________

____

Code Postal ______________________

BUREAU DISTRIBUTEUR ______________________

75008 PARIS 33, Bld des Batignolles. Tél. : 522.70.66. **75009 PARIS** LE PRINTEMPS (Magasin Havre 7e étage) Bld Haussmann. Tél. : 285.22.22 Poste 30.33. **75008 PARIS** 31, Bld des Batignolles. Tél. : 522.70.60. **59000 LILLE** 21 bis, rue de Valmy. Tél. : (20) 57.88.43. **44013 NANTES CEDEX** 21 A, Bld G. Guist'Hau BP 388. Tél. : (40) 47.53.09. **33081 BORDEAUX CEDEX** Immeuble de la Croix du Palais. Rue du Corps Franc Pommiès MERIADECK. Tél. : (56) 96.28.11. **06400 CANNES** 14, Bld de la République. Tél. : (93) 39.29.09.

SIVEA®

TRACE

le magazine des utilisateurs de TRS-80*

TRS-80 modèle 1, TRS-80 modèle 2, TRS-80 modèle 3, TRS-80 couleur, TRS-80 de poche, poquettes Sharp* Vidéo-Génie*, LNW* : si vous utilisez ou si vous comptez acheter un de ces ordinateurs, sachez que la revue TRACE a été créée pour vous. TRACE vous fournit quatre fois par an des programmes, des astuces, de nouvelles idées d'applications. TRACE est indépendant de tout constructeur, et vous tient au courant de toutes les nouveautés dans les domaines des logiciels, des matériels et des périphériques, quelle qu'en soit la source. TRACE teste pour vous, en toute objectivité et indépendance, les produits matériels ou logiciels qui vous intéressent. TRACE n'est pas en vente chez les marchands de journaux. Pour vous abonner ou pour recevoir un numéro, il vous suffit de nous retourner le bon de commande ci-dessous.



*TRS 80, Sharp, Vidéo-Génie et LNW sont des marques déposées.

Trace a le même éditeur qu'ORDI-5 le magazine des utilisateurs de ZX SINCLAIR

TRACE, le complément indispensable de votre TRS-80

BON DE COMMANDE

à retourner à TRACE, 8 rue Saint-Marc 75002 PARIS

TP

Nom ______________________ Profession ______________________

Adresse ____

Pays ______________ Code postal ______________ Ville ______________

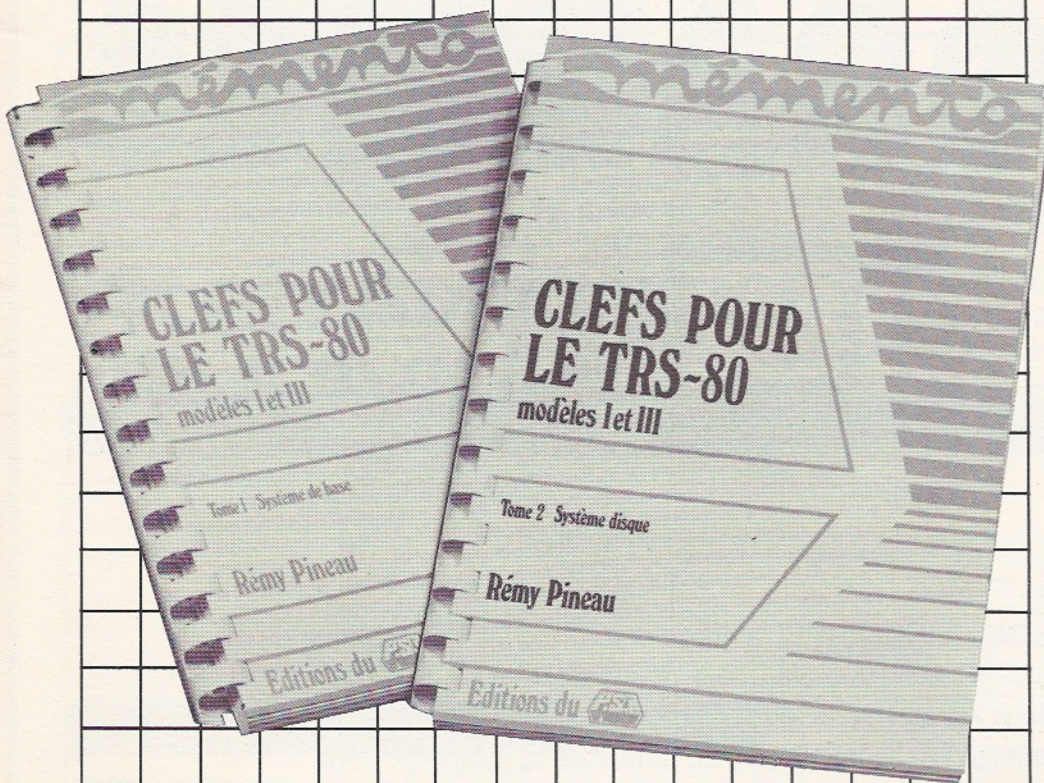
- Je désire recevoir le n° 3 ☐ le n° 4 ☐ le n° 5 ☐ le n° 6 ☐ de TRACE (NB : les nos 1 et 2 sont épuisés). (Prix d'un n° 30 FF; Etranger** 35 FF; par avion 50 FF).
- Je désire recevoir la disquette d'accompagnement des nos 1 et 2 ☐ du n° 3 ☐ du n° 4 ☐ du n° 5 ☐ du n° 6 ☐ (Prix d'une disquette 55 FF; Etranger** 65 FF).
- Je désire m'abonner à TRACE pour 4 nos à partir du n° 3 ☐ du n° 4 ☐ du n° 5 ☐ du n° 6 ☐ du n° 7 ☐
  ☐ avec disquette (Tarif France : 290 FF; Etranger** : 320 FF; par avion 400 FF).
  ☐ sans disquette (Tarif France : 105 FF; Etranger** : 120 FF; par avion 190 FF).

(Actuellement TRACE est trimestriel). Ci-joint mon règlement indispensable par chèque bancaire ☐ chèque postal ☐ virement ☐

** Pour les pays autres que la France, utiliser un virement en FF compte Crédit Lyonnais Paris n° 30002 00402 8401 M. Les frais de virement sont à la charge de l'acheteur.

AGRAPH

LES INDISPENSABLES



Clefs pour le TRS-80 - *Tome 1*
par Rémy Pineau
Débutant ou spécialiste, ce "mémento" vous est destiné. Son but est de vous fournir les "clefs" d'un accès rapide et efficace à l'apprentissage ou à la maîtrise de l'utilisation des Modèles I et III. Ce premier tome aborde essentiellement les caractéristiques de la version de base du TRS-80 (plans, mémoires, brochages, codes, syntaxe, message d'erreurs du Basic, codes machine Z80); cependant les deux derniers chapitres, consacrés aux adresses ROM et RAM du système, servent de transition au 2e tome destiné aux utilisateurs actuels ou potentiels de disque.

192 pages - 112,00 FF / 685,00 FB

Clefs pour le TRS-80 - *Tome 2*
par Rémy Pineau
Possesseurs de TRS-80 modèle I ou III équipé de disques, ce deuxième tome des "clefs" vous est destiné. Il a pour objectif de vous donner les principales informations concernant l'utilisation des disques à travers l'étude des différents formats de stockage de données, des commandes des Systèmes d'Exploitation Standards (TRS-DOS, LDOS, NEWDOS) et des utilitaires associés les plus courants.

200 pages - 112,00 FF / 685,00 FB



P.S.I. DIFFUSION
BP 86 - 77402 Lagny-S/Marne Cedex
FRANCE
Téléphone (6) 006.44.35

P.S.I. BENELUX
5, avenue de la Ferme Rose
1180 Bruxelles
BELGIQUE
Téléphone (2) 345.08.50

au Canada
SCE Inc.
65, avenue Hillside
Montréal (Westmount)
Québec H3Z1W1
Tél. : (514) 935.13.14

en Suisse
CRISPA
Case postale 152
Chemin Ste Agnès/des roches 8A
1701 Fribourg
Tél. : (037) 26.43.43

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg à P.S.I. BENELUX

TRS 3

DESIGNATION	NOMBRE	PRIX
par avion : ajouter 8 FF (75 FB) par livre	TOTAL	

Signature (obligatoire pour paiement par carte de crédit)

☐ Paiement par chèque joint ☐ Paiement en FF par carte bleue VISA (à P.S.I. DIFFUSION uniquement)

VISA

N° ________________ Date d'expiration ____

NOM ________________ PRENOM ________________

rue ________________ n° ____

Code postal ________ Ville ________________

AGRAPH



A propos des disquettes TRACE

Il arrive occasionnellement que certains de nos lecteurs n'arrivent pas à relire correctement nos disquettes (c'est un comble !). Nous avons constaté après retour, que certaines d'entre elles avaient souffert durant l'expédition et se trouvaient aplaties. La conséquence directe est que le disque de mylar tourne avec beaucoup de difficultés dans sa pochette (on l'entend d'ailleurs lors d'une tentative de lecture). Avant de nous la renvoyer pour échange, essayez donc un remède assez simple : il suffit de passer une fine aiguille à tricoter (2 mm de diamètre) sous les 4 bords de la pochette afin de leur redonner leur forme initiale. Il est conseillé toutefois de bien maintenir l'aiguille pres du bord afin de ne pas risquer de rayer le disque ! Certains lecteurs s'obstinent à vouloir lire directement la disquette (DIR) sur modèle 3 avec TRSDOS... C'est peine perdue ! Le transfert ne peut se faire que par l'utilitaire CONVERT ou par un système d'exploitation capable de lire des disquettes en simple et double densité (NEWDOS-80 par exemple). Dans ce dernier cas, il sera probablement nécessaire de signaler au système par la commande adéquate, que la seconde unité est en simple densité. Pour information, nous donnons la commande PDRIVE de NEWDOS-80 permettant l'exploitation directe de nos disquettes (DIR, COPY, etc...) sur modèle 3 :

PDRIVE,0,1,TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2,A.

Déformations... professionnelles !

- Au lancement du programme DEFORME publié dans le numéro 5, j'obtiens une erreur "Illegal fonction call in 2290". Quant à SEGMENT, je ne parviens pas à introduire le "Memory Size" indiqué, l'écran ne me pose à aucun moment la question... De plus, BASPY et DRIVCLA restent assez mystérieux pour moi, du fait de leurs versions multiples sur la disquette. Lequel faut-il choisir ? Pour BASPY, le retour au BASIC ne fonctionne que si la commande B ou Q est suivie d'un BREAK. Est-ce normal ? (Robert Lutz, Strasbourg)

* Oui, c'est un peu notre faute... En fait, DEFORME et SEGMENT font partie du même article et le mode d'emploi qui associe les deux programmes s'est perdu dans la nature... ! Toutes nos excuses. Il faut d'abord lancer SEGMENT pour pouvoir utiliser DEFORME. Quant au "Memory Size" auquel vous faites allusion, vous devez probablement posséder un Vidéo-Génie (vous ne le précisez pas sur votre lettre). Si tel est le cas, à la mise sous tension, votre ordinateur doit afficher "READY ?" et il suffit à ce moment là de frapper la valeur de la taille mémoire et d'appuyer sur la touche ENTER.
En ce qui concerne les versions multiples, seul BASPY/BAS est directement utilisable. BASPY/ASM est le source assembleur (utilisable uniquement si vous désirez apporter des modifications au programme) et /CMD est le code issu de l'assemblage et qui, après chargement, devient /BAS...
Nous espérons que le numéro 6 vous a apporté quelques précisions à ce sujet. De même pour DRIVCLA : /ASM est le source assembleur et /CMD est le code objet directement utilisable sous DOS.
Les commandes B et Q de BASPY ne nécessitent pas l'appui de la touche BREAK car elles provoquent dès leur appui le retour au BASIC exactement à l'endroit où vous l'aviez quitté. Aucun affichage ne se produit dans ce cas. Si vous appelez BASPY (par SHIFT M) alors que BASIC est en train d'afficher un texte (commande LIST, par exemple), le retour au BASIC se poursuivra exactement là où il a été interrompu. Lisez attentivement les dernières lignes de la page 46 du numéro 5 et faites ce qui y est indiqué...

- Dans l'article sur le "driver clavier" des numéros 5 et 6, il est question d'une touche CTRL. Comment l'obtient-on au clavier ? (SAMC, Vaujours).

* Le caractère "flèche basse" a "sauté" dans le texte de la page 51 du numéro 5. Toutes nos excuses pour cet incident. Il fallait lire (colonne gauche) : la fonction contrôle (CTRL) directement en pressant SHIFT et ... "flèche basse", sans déplacement du curseur, comme avec les nouvelles MEM.

Choisir son disque sur modèle 3

- Je possède un TRS modèle 3 et je l'ai équipé moi-même d'un contrôleur et de 2 unités de disquettes. J'utilise le système d'exploitation NEWDOS-80 et j'en suis très satisfait. Par contre, il m'est impossible de faire tourner TRSDOS sur cette configuration. De quoi celà peut-il bien venir ? (Jean-Philippe Maquaire, Galan).

* Pour que TRSDOS puisse fonctionner correctement sur modèle 3, il est nécessaire que vos unités de disquettes soient suffisamment performantes pour accepter des temps "piste à piste" de 5 millisecondes, ce qui semple ne pas être le cas (en général, les Tandon conviennent parfaitement). Par contre, un lecteur connaît peut-être le "patch" à appliquer à TRSDOS pour qu'il accepte des temps supérieurs. Appel au peuple !

... et choisir son assembleur !

- Suite à la remarque de M. Eichaker concernant EDTASM sur cassette modèle 1, il y a effectivement une "bogue" dans le programme qui ne permet de sauver seulement que les 610 premières lignes de code source. La version correcte porte la référence 700-9003 et elle est, en principe, échangée gratuitement contre la version erronnée. (François Heizmann, Brignais).

* Voilà qui va faire plaisir aux nombreux utilisateurs qui nous ont signalé cette anomalie. On nous signale toutefois un autre remède (ci-dessous). Grand merci à ces deux lecteurs.

- Une fois lé programme chargé, il suffit de remplacer l'octet 00 de l'adresse 4AEA par l'octet FF et tout rentre dans l'ordre ! Ceci peut se faire en utilisant un programme moniteur ou encore, EDTSAM étant chargé, de repasser au BASIC (BREAK) et de faire : POKE 19178,255. Pour revenir dans l'assembleur, faire : SYSTEM et /19434. (Serge Cabala, Saone).

Accorder les violons...

- Concernant l'article sur MDX-2 de Henri Frère (n° 6), la spécification technique du MC14412 (modem LSI de Motorola) indique qu'un niveau logique 0 sur la broche 10 sélectionne les fréquences CCITT qui (oh surprise !) sont celles utilisées en France ! A priori, un coup de "cutter" et de fer à souder devrait arranger les choses... En ce qui concerne le transfert modèle 1-modèle 3 (cf courrier des lecteurs du n° 6, page 11), il existe sur ce premier un couple de programmes GETTAPE et GETDISK qui sont acceptés sans problème par le modèle 3. Le seul inconvénient est que le processus est désespérément long ! (François Heizmann, Brignais).

* Voilà qui est interessant. Merci pour ces renseignements utiles que certains de nos lecteurs ne tarderont pas à expérimenter !

TRSDOS, quoi de plus ?

- Dans votre article "NEWDOS-80 version 2, quoi de plus" (cf TRACE numéros 5 et 6), je vous signale que la fonction APPEND existe sur TRSDOS 2.3, contrairement à ce qui est indiqué. J'ai aussi remarqué une fonction de ce DOS non-décrite dans le manuel Tandy : CMD"E". Quel est son rôle précis ? (Eric Bertrem, Caluire).

* Effectivement, APPEND existe sur TRSDOS mais depuis la version 2.3 seulement. CMD"E" permet de connaître la dernière cause d'erreur à la suite d'un problème lors de l'exécution d'un programme BASIC (accès disque notamment).

Sauvegardez vos cassettes sur VGS

- Le programme "sauvegardez vos cassettes" est très intéressant mais inutilisable sur mon VGS EG 3003 dont le magnétophone est intégré. Est-il possible d'utiliser la sortie DIN dans ce programme ? (JC. Chevalier, Rion des Landes).

* Si vous vous reportez au courrier des lecteurs du numéro 5 (page 15), il est indiqué que l'accès à la sortie DIN est possible par l'instruction OUT 254,1 qu'il devrait suffir d'ajouter avant l'exécution du programme.

SUPER quoi ... ?

- Dans le numéro 4 (pages 21,22) vous parlez de SUPERZAP de NEWDOS-80. Possédant la version modèle 3, je n'arrive pas à atteindre cet utilitaire (l'ordinateur répond toujours : file not found). Le manuel qui est dans un anglais très technique, n'a pu m'aider. Quelqu'un aurait-il la solution ? (B. de Stobbeleir, Dilbrek).

* La solution est simple. Il suffit de frapper sous DOS (après le message : DOS READY) le mot : SUPERZAP terminé par un appui sur la touche ENTER... C'est aussi simple que celà ! Si ce programme n'est pas trouvé (ce que semble indiquer le message d'erreur), il y a deux possibilités : ou bien votre disquette ne contenait effectivement pas cet utilitaire (et c'est absolument anormal), ou bien vous l'avez supprimé à la suite d'une mauvaise manipulation. Dans les deux cas, nous vous conseillons d'en parler à votre revendeur.

PROFILE sous NEWDOS-80

- J'éprouve des difficultés à charger le programme PROFILE de Tandy sur NEWDOS-80 version 2. Connaissez-vous une solution à ce problème ?(Jean Dubois, Poulseur).

* La documentation Apparat signale (Zap n° 8) une modification à apporter au module INIT de ce programme. Attendu qu'il n'est pas toujours très facile d'obtenir ces informations auprès des revendeurs européens, nous la donnons ci-dessous. Nous vous conseillons aussi de "purger" votre disquette système de façon à laisser suffisamment de place pour le bon déroulement du programme (NEWDOS-80 occupe davantage de place sur la disquette que TRSDOS). ATTENTION : la modification peut varier selon la version de PROFILE. Lisez d'abord le contenu avant d'appliquer l'un des Zaps ci-dessous :

```
INIT, secteur relatif 0, octet 54 :
Avant : 44 38 09 28 0F FE 1A 28 13 C3 3F 52 CD
Après : 44 20 19 00 00 00 00 00 00 00 30 08 CD
```

Ou bien :

```
INIT, secteur relatif 0, octet 63 ou 64 :
Avant : 44 38 11 28 07 FE 1A 28 13 C3 47 52 CD
Après : 44 20 19 00 00 00 00 00 00 00 38 08 CD
```

Toujours au sujet de PROFILE, Gérard Manel (Vaucouleurs) nous signale que le Zap dont il était question dans notre numéro 4 (page 22) concerne une ancienne version. Sur PROFILE V3.0, la même modification doit se faire au secteur 6 octet FE. Merci !

Vous avez dit INPUT ?

- Lorsqu'on exécute l'instruction INPUT, un point d'interrogation est toujours affiché. Existe-t-il un moyen pour le supprimer ou le remplacer par un autre signe ? (Paul Freyss, Illzach).

* Cette instruction étant située en mémoire morte, il n'est pas possible d'en modifier l'exécution. Par contre, si vous possédez le BASIC disque, l'instruction LINE INPUT devrait vous apporter entière satisfaction. Le cas échéant, un petit sous-programme basé sur la fonction INKEY$ vous aidera à résoudre votre problème.

EDTASM+ sur modèle 3 ?

- Poussé par la curiosité qu'avait déclenchée l'article sur l'Editeur Assembleur Plus de Microsoft, je me suis procuré cette cassette. Malheureusement, elle ne tourne pas sur modèle 3. Connaissez-vous les modifications à apporter ? (JC. Draux, Maubeuge).

* Il est tout de même curieux que Microsoft ne soit pas en mesure de fournir une version adaptée au modèle 3. Des lecteurs auraient-ils résolu le problème ?

Rien qu'un p'tit trou...

- Suite à votre article "doubleur de disquettes" du numéro 4, je vous signale que j'utilise depuis plus de 2 ans la "Perforette" à un trou fabriquée en France par CONTE (référence 6011). Pour 10 F, ça marche merveilleusement bien ! (Raymond Debruyne, Ieper).

* Merci pour cette information. Combien de trous en 2 ans ?

Imprimante... peuchère !

- Dans le Figaro Magazine du 11 juin, une publicité Tandy faisait mention de prix particulièrement alléchants sur la gamme d'imprimantes VI, VII et VIII (1150 F pour la LP VII), offre valable sur les stands Micro-Expo et centres mentionnés pendant la durée du 14 au 18 Juin. Bien entendu, j'étais au rendez-vous dès le premier jour de l'offre pour découvrir que, renseignements pris, l'imprimante en question n'était (déjà ?) plus disponible et que l'on pouvait me proposer un modèle de la nouvelle gamme à un prix naturellement plus élevé... Je passe sur les évènements qui ont suivi (qui m'ont coûté une bonne dose d'énervements et de kilomètres à travers Paris) pour signaler que j'ai tout de même obtenu satisfaction. Mais à quel prix, finalement ? Tandy qui semble attacher une si grande importance à la légalité (notamment suite à son action auprès du Club Microtel), a-t-il agit ici de manière vraiment légale ? (R. Joffre, Paris).

* Tandy semble avoir montré dans cette affaire quelques problèmes quant aux communications internes, puisque peu de boutiques avaient l'air informées de la promotion décidée par Tandy-France. Heureusement, vous avez finalement obtenu satisfaction.

Le modèle 1 ? Quel modèle 1 ?

- A l'occasion d'une révision de mon modèle 1, j'ai demandé l'installation du kit minuscules et du générateur de caractères accentués. On m'a répondu que bien que ces accessoires figurent encore au catalogue 83, ils n'étaient plus vendus (!). Peut-être serait-il bon de le signaler à vos lecteurs (Pierre Derrey, La Varenne St Hilaire).

* Voilà qui est fait sans autre commentaire.

Question de Copyright...

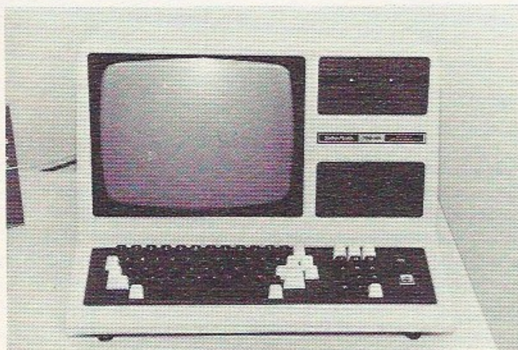
- Comment doit-on faire pour passer un Copyright sur un programme ? (Gérard Naudin, Mézieres).

* Nous avons beaucoup tardé à publier cette question, espérant un jour pouvoir lui apporter une réponse valable ! En fait, il ne semble pas que la législation française soit encore très développée sur ce sujet. Il y a toutefois lieu de distinguer le Copyright associé à la liste d'un programme publié dans un livre ou une revue de de celui écrit magnétiquement sur une cassette ou une disquette.
Un lecteur versé dans ces questions nous apportera-t-il une réponse ?

MAGAZINE

TRACE et ORDI-5 ont une petite soeur ! Son nom est : ORDI-MAGAZINE ! Il s'agit d'une revue trimestrielle consacrée à l'ordinateur PC d'IBM et à ceux qui lui sont compatibles au travers des systèmes d'exploitation MS-DOS et CP/M-86 (ordinateurs 16 bits).

Le TRS modèle 100, dont nous avions annoncé le lancement aux Etats-Unis dans notre précédent numéro est disponible en France. Distribué dans 110 points de vente, le "mini-poste de travail pour cadres supérieurs" (!), ainsi que le nomme son constructeur, est vendu au prix de 5995 F pour la version 8 K octets, et 7495 F pour la version 24 K.



TRS modèle 4

Voici maintenant deux autres nouveautés : le TRS-80 PC-4 et le TRS-80 modèle 12.

Le premier est un ordinateur de poche, compact et économique, dont le prix est de 695 F. Il mesure 0,95 x 16,5 x 7 cm et peut se glisser sans la moindre difficulté dans un porte-document ou un sac de taille moyenne. Le clavier se compose de 53 touches avec bloc de saisie numérique de 10 touches. Son affichage à cristaux liquides de 12 caractères permet le défilement horizontal (62 caractères) et l'affichage des caractères minuscules. Son langage est le BASIC qui comprend 23 commandes, 15 fonctions, un mode EDIT (édition) et un mode DEBUG (mise au point). Les valeurs numériques ont une longueur de 10 chiffres significatifs et les chaînes 30 caractères. Le PC-4 peut stocker jusqu'à 10 programmes de faible dimension dont le lancement est associé à 10 touches du clavier. Le fonctionnement est assuré par deux piles au lithium dont la longévité est prolongée par un dispositif de mise hors tension automatique en cas de non-utilisation. La machine est livrée avec un mode d'emploi complet et un guide programmation (en français, nous l'espérons). Les extensions prévues sont : un module mémoire 1 K octets (139 F) qui fait passer la mémoire de 544 à 1568 "pas" ; une interface cassette (349 F) à 300 bauds qui se connecte à l'arrière du PC-4, alimentée par 2 piles "crayon" ; une imprimante (749 F) qui imprime des lignes de 20 caractères à la vitesse de 60 lignes par minute (matrice de 5 x 7 points). Elle fonctionne sur batteries rechargeables au cadmium-nickel est est livrée avec un adaptateur/ chargeur se raccordant au secteur 220 V. Avec ses périphériques (interface cassette et imprimante), le PC-4 forme un tout compact de 3,6 x 17,2 x 17,8 cm.

Le modèle 12 est d'un tout autre encombrement puisqu'il est d'une taille voisine de celle du modèle 2. Ce nouveau matériel offre aux utilisateurs une très grande facilité d'extension, des possibilités de programmation en divers langages et une capacité de stockage accrue. Il représente une solution économique pour les applications professionnelles nécessitant des fichiers de données volumineux ou de nombreuses transactions, y compris le contrôle d'inventaire, les dossiers médicaux, la comptabilité, etc... La configuration avec clavier AZERTY et une unité de disquette est vendue 32.995 F (39.995 F avec 2 unités). Il est équipé d'un microprocesseur 8 bits Z-80 A (version rapide) qui, couplé à un contrôleur d'accès direct à la mémoire (DMA) et à un système d'interruption vectorielle, assure un débit d'informations plus rapide. La capacité mémoire est de 80 K octets pour la mémoire vive et 1,25 millions d'octets par unité de disquette. Le moniteur vidéo du modèle 12 se caractérise par un écran vert haute résolution de 30 cm, affichant 24 lignes de 40 ou de 80 caractères (majuscules et minuscules, plus 32 caractères graphiques). Le clavier amovible est muni de 82 touches dont un bloc de saisie numérique et 8 touches de fonctions.

Les extensions sont possibles grâce à 6 connecteurs d'interface permettant de recevoir des cartes additionnelles (extension mémoire jusqu'à 144 K, graphique haute résolution, kit d'émulation du modèle 16 donnant des capacités de traitement 16/32 bits, avec une augmentation de la mémoire jusqu'à 768 K). Le modèle 12 est équipé en standard d'une interface parallèle, de 2 interfaces série RS-232 et d'un accès pour disques externes. Il est compatible avec tous les logiciels du modèle 2. Le nouveau système d'exploitation TRSDOS accepte les disquettes 8 pouces en double densité, simple ou double face, mais aussi les nouvelles unités de disques durs 12 millions d'octets. Ces dernières (34.995 F pour le disque maître et 24.995 F pour le disque esclave) présentent une capacité de stockage accrue ainsi qu'un temps d'accès rapide grâce à la technologie Winchester. L'unité se compose d'un lecteur de disques 5 pouces 1/4 isolé hermétiquement de l'environnement extérieur et incluant un système de recyclage d'air filtré. Le débit est de 600 Ko par seconde et le temps d'accès piste à piste est de 3 millisecondes (8,3 ms en moyenne). Quatre disques durs peuvent être raccordés à l'ordinateur.

(Tandy France, 211-213 Bd Mac Donald, 75019 PARIS)

A l'occasion du Sicob, les éditions P.S.I. annoncent 27 titres de livres et 7 disquettes d'accompagnement. Parmi les ouvrages susceptibles d'interesser les lecteurs de TRACE, nous citerons : Méthodes de calcul numérique, tome 2 (Claude Nowakoxski), Programmer en Forth (Alain Pinaud - C'est un ami ! NDLR), le BASIC Microsoft (Ken Knecht), Techniques de programmation des Jeux (David Levy), BASIC pour tous (Jacques Boisgontier et Sophie Brébion), Boîte à outils pour PC-1500 (JP Lhoir et Richard Poret), La découverte du PC-1251 (Jean-Pierre Richard), Clefs pour le TRS-80, tomes 1 et 2 (Rémy Pineau), Bibliothèque scientifique en Pascal (Hervé Haut).

(Editions du PSI, Parc Industriel Nord, Bât 9, 77200 TORCY).

SIVEA annonce l'ouverture de nouvelles boutiques à Paris, Bordeaux et Cannes. Ces ouvertures ont eu lieu durant les mois de Juin, Juillet et Août, et "correspondent à la volonté de SIVEA de décentraliser et d'améliorer la qualité des services rendus à la clien-

tèle". Avec ces nouvelles extentsions, SIVEA dispose aujourd'hui de 8 points de vente.

SIVEA Bordeaux : Immeuble de la Croix du Palais
Rue du Corps Franc Pommiés
33081 BORDEAUX CEDEX

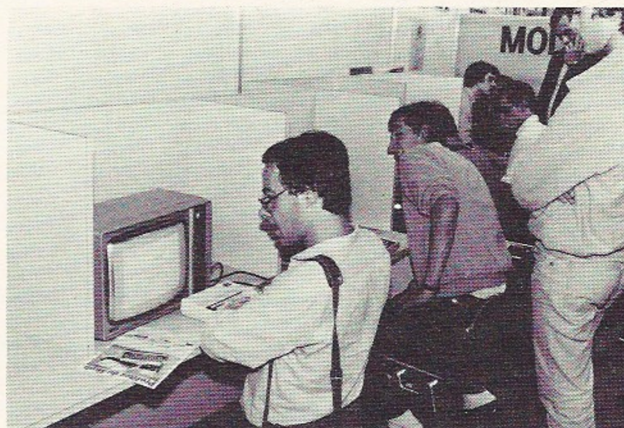
SIVEA Cannes : 14 boulevard de la République
06400 CANNES

A Paris, la boutique du 31 Boulevard des Batignolles est désormais exclusivement consacrée à l'informatique pour l'entreprise (on peut y voir le PC d'IBM, Lisa d'Apple, Victor, Apple 3...) et à quelques mètres de là (au 33 du même boulevard), une nouvelle boutique est consacrée à l'informatique domestique et aux loisirs informatiques. Encore un peu plus loin (33 rue de Moscou), le service après-vente SIVEA assure la maintenance de la plupart des matériels d'informatique individuelle et peut réaliser à la demande la plupart des câbles d'interface pour connecter les différents éléments de systèmes d'ordinateurs.

LES SURPRISES DU SICOB...

Tandy (que de surprises actuellement de la part de ce constructeur !), annonçait la sortie de 2 nouveautés au Sicob : le modèle 4 (dont nous avions annoncé la commercialisation aux Etats-Unis dans notre numéro 6) et le MC-10.

Le modèle 4, désormais fabriqué en France par MTE (Matra Tandy) est vendu 18500 F ttc pour une configuration 64 K, 2 unités de disquettes, interfaces parallele et série (9000 F en 16 K cassette). Rappelons brièvement ses caractéristiques (voir TRACE n° 6 pour plus de détails) : carrosserie du modèle 3 (mais de couleur blanche), microprocesseur Z-80 rapide (4 Mhz), 64 à 128 Ko de mémoire vive, clavier AZERTY, écran de 24 lignes de 80 caractères, systèmes d'exploitation : TRSDOS 6.0 ou CP/M Plus.



Des Fanas autour d'Alice

Le MC-10 quant à lui, est un ordinateur à usage domestique, principalement destiné à l'initiation. Il est distribué par Tandy sous le nom MC-10 (couleur blanche) et par Matra sous le nom "Alice" (couleur rouge). Son prix est de 1200 F. D'une taille de 22 x 18 x 5 cm, il se relie aux téléviseurs couleurs munis d'une prise Péritel et offre 8 couleurs sur un écran de 16 lignes de 32 caractères. Son clavier de 48 touches est de type AZERTY. Il permet l'affichage des commandes BASIC sur un seul appui de touche (Control) et de 16 codes graphiques (Shift). Son microprocesseur est un 6803 et sa capacité mémoire est de 4 K octets. Toutefois, un module d'extension de 16 K peut lui être ajouté. Le BASIC résident est naturellement signé Microsoft et est pourvu de commandes sonores. Le MC-10 possède deux interfaces de communication : l'une est destinée à un cassettophone et l'autre de type série, à une imprimante ou un modem. Dans cette version - et compte tenu du prix -, les possiblités sont assez restreintes mais sont toutefois suffisantes pour l'initiation. A quand maintenant, l'annonce du BASIC étendu ... ?

Dernière nouvelle : le manuel sur le langage machine du PC-1500 doit paraître d'ici la fin de l'année...!

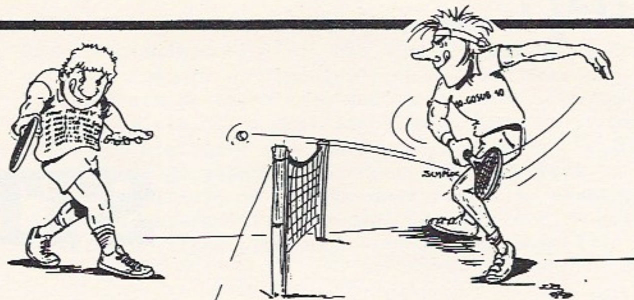
Une nouvelle extension de mémoire C/MOS de 26 K RAM pour PC-1500:PC2 est maintenant disponible. Elle porte la capacité totale à 29.296 octets accessibles pour la programmation en BASIC ou en langage machine. Ces cartes sont placées dans la machine par les soins de l'importateur.

(L.L. Informatique SPRL, Bd Paul Janson, 82, B-6000 CHARLEROI (Belgique)).

Le club belge Micro-GDV compte actuellement 422 membres (35 % de radio-amateurs) dont 202 sont des utilisateurs de TRS-80.

(Micro-GDV, rue de l'Eglise, 54, B-4710 HERBESTHAL-LONTZEN)

Côté court



UN INKEY$ PATIENT !

La fonction INKEY$ du BASIC est très utile pour la saisie "au vol" de caractères issus du clavier. Pourtant, on aboutit toujours à la boucle classique :

```
10 A$=INKEY$ : IF A$="" THEN 10
```

car cette fonction n'attend pas la frappe du caractère. Voici une routine d'une ligne qui permettra l'attente d'une frappe au clavier. Elle fait appel au mot-clé NAME inutilisé en BASIC cassette. Avant son emploi, pensez à spécifier un MEMORY SIZE (32000 dans notre exemple). Avec cette nouvelle commande, vous possédez maintenant une fonction INKEY$ + l'attente d'une touche au clavier + la gestion possible de la touche BREAK + une économie d'octets... Voici d'abord un exemple d'appel (on retrouve le caractère dans A$). Notez le test du BREAK par CHR$(1) :

```
10 GOSUB 50000
100 NAME : IF A$=CHR$(1) THEN END
110 PRINT A$ ;
120 GOTO 100
```

Et voici la commande NAME :

```
50000 A$="*":K=32000:FORI=KTOK+16:READR:POKEI,R:NEXT:
I=16783:POKEI,K-INT(K/256)*256:POKEI+1,K/256:RETURN:
DATA217,205,73,0,42,249,64,17,4,0,25,94,35,86,18,217,201:
REM3B
```

Rémi Piette

FLASH...

Cette courte séquence peut être ajoutée dans un programme pour produire des "flashes" sur l'écran sans détruire le texte affiché (pour un nombre pair d'appels) :

```
10 A%(0)=33 : A%(1)=32316 : A%(2)=16366 : A%(3)=9079 : A%(
4)=29899 : A%(5)=-2264 : A%(6)=201 : DEFUSR=VARPTR(A%(0))
```

Exemple d'appel :

```
20 FOR N=1 TO 10 : B=USR(0) : NEXT
```

En configuration cassette, remplacer DEFUSR=... par :

```
A=VARPTR(A%(0)) : POKE 16526,(A AND 255) : POKE 16527,A/256
```

Claude Ferrand

JEU DE TAB...

Regardez la ligne suivante. Elle semble bien banale ? C'est vrai, et pourtant il suffit de changer la longueur de la chaîne de "I" ou de supprimer la virgule (avant cette chaîne) pour obtenir des effets très différents...

```
1 FOR I=0 TO 255 : PRINT TAB(I) , "IIII" ; : NEXT :
GOTO 1
```

Jean-Paul Calvier

"SPEEDY GONZALES'S AUTO-REPEAT"...

Certains programmes ne servent apparamment à rien ou attendent de trouver une utilisation (en général, ce ne sont pas les plus simples à écrire !). En voici un exemple :

```
5 CLS
10 PRINT CHR$(PEEK(16537)); : GOTO 10
```

Faire RUN et appuyer sur quelques touches... Essayer aussi la touche "flèche arrière" ou la touche CLEAR.

Un mot d'explication. L'adresse 16537 est utilisée par la fonction INKEY$ et contient le dernier caractère introduit au clavier. Bien que le programme ci-dessus n'utilise pas cette fonction, l'interpréteur effectue une lecture systématique du clavier au début de chaque ligne BASIC et met à jour cet emplacement mémoire.

Benoît Gilon

UNE UTILISATION PEU COURANTE DE VARPTR...

Regardez le programme ci-dessous. Que peut-il bien faire d'après vous ? Alors frappez-le, faites LIST et RUN (la boucle FOR/NEXT permet de mieux voir ce qui se passe). En y regardant d'un peu plus près, nous voyons que les 3 POKE et le VARPTR déclarent une chaîne de caractères DANS L'ECRAN plutôt qu'en mémoire, ce qui est peu banal !

```
10 PRINT @960,"*****" : A$=" " : X=VARPTR(A$) : POKE
   X,64 : POKE X+1,0 : POKE X+2,60
20 PRINT A$; : FOR X=1 TO 50 : NEXT : GOTO 20
```

Une version plus lente - mais humoristique - consiste à remplacer la ligne 20 par :

```
20 FOR X=64 TO 1 STEP -1 : PRINT MID$(A$,X,1); :
NEXT : GOTO 20
```

Fernand Gilles

STEP -1

Cette nouvelle rubrique est destinée (comme son nom évocateur l'indique aux programmeurs BASIC que vous êtes !) à apporter des précisions, des améliorations ou à corriger des erreurs concernant les articles ou programmes des numéros précédents. Elle est née et sera alimentée en grande partie par vos lettres. Aussi, ne vous en privez pas ! Merci.

COTE COURT / TRACE n° 5, page 16

- Une erreur de frappe (ordinaire) s'est glissée dans la ligne du programme "super-hexa-ordinaire"... Il faut ajouter une seconde fermeture de parenthèse avant "-48".

DRIVER CLAVIER / TRACE n° 5

- Le programme DRIVCLA5/ASM figurant sur la disquette ne peut pas être traité par EDTASM version disque. La modification suivante (réalisable avec un "patcher" disque style SUPERZAP) m'a permis de résoudre ce petit problème. (Ndlr : il doit s'agir d'une ancienne version d'EDTASM car nous n'avons pas constaté cette anomalie).

Piste 10, secteur 5, octets 0 à 13 :

```
AVANT : BOBO BOB1 B020 3B20 5355 5045 5220 4452...
APRES : D344 5249 5643 35BO BOBO B1BO 203B 4452...
```

D. Lesage

TRAITEMENT DE TEXTE EN BASIC / TRACE n° 5 et 6

- Avec les modifications ci-dessous, il est possible de concaténer deux ou plusieurs textes dont les morceaux sont lus dans l'ordre logique sur cassette ou disquette.

- ligne 680 : remplacer I=-2 par I=M-2
- ligne 530 : remplacer I=0 par I=M

De plus, il est bon après la lecture d'une page, de se retrouver dans la même configuration qu'à l'écriture de cette page. Aussi, peut-on ajouter les lignes :

```
705 INPUT#-1,JL,MA
785 PRINT#-1,JL,MA
535 INPUT#1,JL,MA
605 PRINT#1,JL,MA
```

G. Mignard

ROULETTE / TRACE n° 4

- Bravo à Frédéric Mora pour son superbe programme de roulette. Il y a toutefois une petite imperfection qui se produit lorsqu'un joueur décide de miser tout son argent sur le premier coup. Dans ce cas, R(I,K) (objet des mises) et G(I,K) (montant des mises) du tour précédent ne sont pas annulés. La ligne 1810 devient donc :

```
1810 IF F(I)=0 THEN 1900
```

Allez, Messieurs, plus de tricherie possible : à vos claviers et faites vos jeux !

Guy Berger

DRIVER CLAVIER / TRACE n° 5 et 6

- Ce programme est un très bel outil qui possède cependant une petite erreur. Lancez en effet le programme BASIC : 10 GOTO 10... puis pressez simultanément sur CLEAR et ENTER pour appeler l'éditeur de texte. En essayant de déplacer le curseur, on plante la machine. Le remède consiste à ajouter dans le programme :

```
535     POP     AF
```

Après cela, le programme ne se plante plus mais le déplacement du curseur laisse une trace. Cette nouvelle erreur est corrigée par :

```
202     LD      HL,(4020H)
204     LD      A,(HL)
206     LD      (4022H),A
```

L'avantage de cette modification est de pouvoir sauter à cet éditeur à partir de la fonction USR par exemple, sans perturber l'affichage vidéo, puis de retourner au programme en pressant la touche BREAK. L'adresse du saut s'effectue alors à la ligne 202 (& HFD88 en version disquette).

Serge Cabala

TAQUIN / TRACE n° 6

- Bravo à l'auteur : j'ai beaucoup apprécié l'idée, mais je le soupçonne d'avoir volontairement joué un bon tour aux lecteurs ! En effet, tel que le programme se présente, la fin du jeu conduit à deux configurations :

```
 1  2  3  4
 5  6  7  8
 9 10 11 12
13 14 15
```

qui est la solution cherchée, mais aussi :

```
 1  2  3  4
 5  6  7  8
 9 10 11 12
13 15 14
```

configuration dans laquelle il est impossible de gagner !

Pour les "matheux", c'est une simple question de permutation "paire" ou "impaire". Je propose donc une modification de façon à ce que la solution soit toujours possible. Il s'agit simplement, lors de la distribution aléatoire des cases, de calculer la "signature" de la permutation. Si cette dernière vaut -1 (cas impossible), on inverse deux cases. Au passage, je modifie également le procédé de distribution afin d'éviter de tirer plusieurs fois le même nombre !

Voici les lignes à modifier :

```
 20 DEFINT C-T : DIM T(4,4),R(16),S(15)
260 FOR I=1 TO 15 : S(I)=I : NEXT I
270 J=16 : FOR I=1 TO 15
280 J=J-1 : K=RND(J) : R(I)=S(K) : S(K)=S(J)
290 NEXT I
300 K=1 : FOR I=2 TO 15 : FOR J=1 TO I-1
310 K=K*SGN(R(I)-R(J))
320 NEXT J,I
330 IF K=-1 THEN R=R(1) : R(1)=R(2) : R(2)=R
```

Gérard Chevin

Trucs à Brac



BUFFER "BIDON"

Il est souvent pratique de pouvoir disposer en BASIC d'un emplacement memoire ne gênant personne... pour y placer une routine binaire, par exemple. Les utilisateurs de systèmes à disquettes savent que le système réserve en mémoire autant de "buffers" (ou zones tampon) que de fichiers pouvant être ouverts simultanément (réponse à HOW MANY FILES sous TRSDOS). La valeur implicite est de 3. Ainsi, il est possible d'utiliser l'un de ces buffers pour y stocker 256 octets de données. Son adresse est obtenue ainsi :

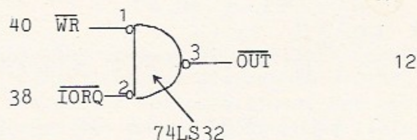
```
10 I=1    ' numero du buffer
20 OPEN "R",I,"BIDON" : FIELD I,1 AS A$
30 V=VARPTR(A$) : CLOSE : KILL "BIDON"
40 PRINT "Le debut du buffer";I;"se trouve a l'adresse";
   PEEK(V+1)+256*PEEK(V+2)
```

Roger Brousmiche

UN VGS "SYNPAT"...

Suite à l'article sur le synthétiseur vocal "SYNPAT" de Pentasonic, voici l'adaptation qui fonctionne très bien sur le Vidéo Génie.

Broche sortie bus VGS	Signal	Broche carte SYNPAT
15	D0	30
14	D1	22
12	D2	32
16	D3	26
20	D4	18
11	D5	28
18	D6	24
17	D7	20
1,2,49,50	masse	8



Le circuit 74LS32 sera alimenté par le VGS en faisant les liaisons :

Broche 19 VGS (+5 V) ---- patte 14 LS32
Broche 50 VGS (masse) --- patte 7 LS32

Dominique Brioux

C'EST LA TETE, DOCTEUR !

Après une série de problèmes pour lire les cassettes sur mon modèle 1, j'ai réussi à obtenir un résultat significatif en retouchant l'azimut de la tête de lecture du magnétophone (alignement par rapport à la bande). Il faut pour cela, passer un petit tournevis dans le trou se trouvant au-dessus du I de REALISTIC et de tourner jusqu'à obtenir le son le plus aigu. Depuis, je ne prends presque plus de crises de nerfs !

Jean-Paul Favier

NE PARTEZ PLUS A L'AVENTURE SANS DICTIONNAIRE !

Si comme moi, vous avez un certain penchant pour les jeux de rôles et pour les quêtes de trésor et si vous aimez les énigmes, alors vous possédez sûrement quelques jeux de la série ADVENTURE. Toutefois, un problème risque de se poser assez rapidement : l'impossibilité de progresser dans le jeu. Trois solutions s'offrent à vous : consulter des amis qui sont peut-être allés plus loin que vous ; reprendre à zéro en notant tous les éléments connus et les obstacles, au prix de plusieurs jours de "sèche" ; écrire à Adventure International pour recevoir les "hint-sheets" de la série (mais cela ne résoudra pas tous les problèmes). L'une ou l'autre de ces solutions arrive presque toujours à l'impasse. Mais ne jetez pas votre jeu en disant "c'est impossible ! Ce jeu est idiot !" car je vous propose de règler rapidement une partie de vos problèmes en vous permettant d'avoir tous les éléments en mains.
Il faut savoir que chaque jeu d'aventure possède son propre vocabulaire (verbes et noms) et souvent, le problème est que nous ne savons pas quel mot employer. Avec les éléments qui suivent, la totalité du vocabulaire apparaîtra sur l'écran...
Il y a deux types de procédure parce que deux séries différentes : la série ADVENTURE classique et la série OTHER VENTURE. Voyons d'abord la première. Après avoir chargé le programme par la commande SYSTEM, par exemple :

```
>SYSTEM
*? ADVENT
```

appuyer sur BREAK et taper :

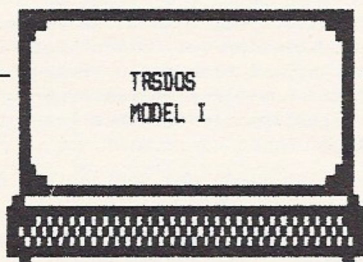
```
FOR I=18000 TO 38000 : PRINT CHR$(PEEK(I)); : NEXT
```

Vous verrez alors défiler le contenu de la mémoire sous forme ASCII (arrêt momentané par SHIFT @). Ce type de programme comporte un vocabulaire formé de mots de 3 ou 4 lettres chacun. Il faut ajouter à cela les commentaires descriptifs et les messages d'erreurs. Prenez des notes passage !

Avec la série OTHER VENTURE, charger le programme comme précédemment mais au lieu de faire BREAK, appuyer sur le bouton RESET. Autant le dire : ça ne marche pas toujours car ces programmes sont dotés d'un "auto-start"... Il faut parfois "réveiller" la machine en passant un objet métallique sur les contacts de la prise d'interface placée à l'arrière du clavier (cela peut paraître dangereux mais c'est sans conséquences fâcheuses !). En général, on obtient ensuite le message MEM SIZE?. Appuyer alors sur RESET puis sur BREAK, et taper la ligne de programme ci-dessus pour lister la mémoire. En cas de plantage, remplacer 18000 par 17000 ou faire un CLEAR 50 et refrapper la ligne. Parfois, un NEW est nécessaire (sans dommage pour la mémoire programme). Cette seconde partie possède un vocabulaire plus restreint sous une forme différente de la première. Par exemple : BRBE SOLD représente "BRIBE SOLDIER", soit deux mots. Par contre, les commentaires sont plus détaillés.
Comme vous le voyez, la première série est plus simple à forcer mais la seconde est plus intéressante du point de vue du jeu. A vrai dire, je n'ai jamais réussi à terminer sans aide un "Other Venture". Bonne chance, et ne désespérez pas : on peut toujours y arriver !

Philippe Mauriras

Kit et double !




```
TRS-80 MODEL I TRSDOS VERSION 2.7DD  THU MAY 20, 1982
48K SYSTEM, NUMBER OF DRIVES = 2   SERIAL #: 5208200009
COPYRIGHT 1981 TANDY CORP.  ALL RIGHTS RESERVED.
UNAUTHORIZED REPRODUCTION OF THIS SOFTWARE IS PROHIBITED
AND IS IN VIOLATION OF UNITED STATES COPYRIGHT LAWS.
AUTO FUNCTION ENGAGED
```

J'ai acquis tout récemment le kit double densité pour modèle 1 de Tandy, disponible depuis peu en Belgique. Je l'ai payé 8.995 F belges (environ 1350 F français).

Le kit comprend un circuit imprimé à implanter dans l'interface d'expansion (aucune soudure n'est nécessaire), une disquette TRSDOS 2.7DD, un luxueux manuel de 176 pages (en anglais), et un dépliant cartonné résumant les commandes du DOS, les instructions DISK-BASIC, et bien d'autres choses encore, dont 14 adresses de sous-routines en ROM.

Le manuel, très complet, décrit en outre 20 sous-routines du TRSDOS. Leurs adresses sont les mêmes que celles du TRSDOS 2.3, du moins pour celles déjà décrites dans le manuel de ce dernier.

Le circuit imprimé contient un contrôleur double densité, un socket dans lequel on enfiche l'ancien contrôleur, plus une dizaine de boîtiers plus petits.
Je n'ai pas essayé d'en relever le schéma, mais il m'a été certifié que le circuit contient l'indispensable séparateur données/horloge.
Le tout s'enfiche dans le socket de l'ancien contrôleur.

La modification doit, en principe, être effectuée dans un atelier Tandy, sinon la garantie est perdue...

Ceci fait, la machine fonctionne exactement comme avant, avec une ancienne disquette dans le drive Ø.
Par contre, dès que l'on y introduit le TRSDOS 2.7DD, notre vieille Thérèse est miraculeusement rajeunie de plusieurs années !

La capacité des disquettes est augmentée de 8Ø %, soit 152 K formatés sur 35 pistes ou 175 K sur 40 pistes.
Le système complet avec ses utilitaires occupe 64 K, mais il est possible de créer un MINI-DOS de seulement 20 K.

On peut accéder aux disquettes simple densité par les seules commandes DIR et COPY.
COPY est devenu nettement plus performant. Il suffit par exemple de faire :

```
COPY /BAS :1 :2
```

pour recopier sur le drive 2 tous les fichiers du drive 1 comportant l'extension BAS, les disquettes 1 et 2 étant indifféremment formatées en simple ou double densité.

La copie fichier par fichier est également possible si l'on ne dispose que d'un seul drive.
Le DOS contient un utilitaire bien protégé nommé ULC qui se charge automatiquement en mémoire haute... si le kit "minuscules" de Tandy est installé, ... sinon il faut se tirer d'affaire autrement.

En désassemblant le haut de la mémoire, j'y ai trouvé bien cachée à l'adresse FEE1H, une routine identique octet pour octet, à celle que j'utilise depuis toujours, et il m'a suffi de faire :

```
POKE 16414, 225 : POKE 16415, 254
```

pour retrouver 'mes' minuscules (en SHIFT).

Vous l'avez déjà compris, le kit double densité confère au modèle 1 la plupart des avantages du modèle 3.

La première page d'écran est quasiment celle du modèle 3, avec bien sûr le curseur clignotant et les touches à répétition.

La bibliothèque du DOS contient 45 commandes, contre 19 pour la version 2.3 ! - Notons au passage :

BUILD = construire un fichier de commandes qui sera exécuté par DO et mis en AUTO pour chaîner une suite d'ordres incluant l'accès au BASIC jusqu'au lancement d'un programme. On peut y inclure une PAUSE qui affiche à peu près 'ENTER pour la suite' après un éventuel message personnalisé.

CREATE = créer un fichier en définissant le nombre d'enregistrements et leur longueur (de 1 à 256 octets.- c'est nouveau !).

DUAL = affichage simultané sur l'écran et l'imprimante.

CONFIG = configurer un drive (nombre de pistes = 35 ou 40, durée d'un changement de piste = 6 à 30 ms).- On ne peut cependant configurer en 40 pistes que les drives initialement prévus pour accepter 40 pistes.

PATCH = pour modifier le contenu d'un fichier.

PURGE = élimination globale de certains types de fichiers.

LPC pour configurer certaines imprimantes, et SPOOL pour créer un buffer de texte permettant à l'imprimante de fonctionner seule, sans perturber l'exécution du programme en cours.

SETCOM = pour définir les paramètres de l'interface série RS-232.

Certaines commandes comme DIR, FREE et LIST acceptent l'option PRT = écriture simultanée sur l'écran et l'imprimante.

etc, etc...

Le DISK-BASIC n'est pas en reste. Le nombre des CMD passe de 4 à 16 ! Par exemple :

CMD'D:1' affiche le DIR du drive 1 avec retour au BASIC.
CMD'C' comprime un programme en éliminant les REM et (ou) les espaces.
CMD'L' charge un fichier binaire.

CMD'Y' donne les références de mots-clés, de variables ou de chaînes.
CMD'Z' pour l'affichage simultané sur l'écran et l'imprimante.
CMD'O' , 1000, F$(1) trie les 1000 premiers éléments de F$() en 7 secondes !

On peut renuméroter les lignes par NAME.

OPEN'E' ouvre un fichier séquentiel pour l'écriture à la fin d'un fichier existant (sans détruire celui-ci).

OPEN'R',2, 'FICHIER:1' , 70 ... et la longueur des enregistrements logiques est de 70 octets !

<.> ou <,> en début de ligne LISTe ou EDITe la ligne courante. De même, flèche haute ou basse LISTe la ligne précédente ou suivante.
L440 ou E440, LISTe ou EDITe la ligne 440 etc... - Il y a maintenant 12 abréviations au service du programmeur pressé.

Tout ceci est positif. De plus la compatibilité semble assurée avec le TRSDOS 2.3, sauf peut être pour certains CALL, PEEK ou POKE par trop spécifiques.

Par contre, des doutes subsistent en ce qui concerne l'utilisation inconditionnelle des progiciels écrits pour le modèle 1, et surtout le manuel ne dit mot des possibilités d'accès aux disquettes du modèle 3 !

Côté négatif, j'ai seulement relevé (jusqu'à présent), la difficulté de récupération d'un programme BASIC par BASIC * (retour au BASIC après un court voyage dans le DOS).
D'après mes investigations, avant de taper CMD'S' ou CMD'I' il faut faire CLEAR 50, mais le programme est quand même perdu après un RESET, volontaire ou non !

La taille de la mémoire vive disponible est aussi raccourcie d'environ 800 octets. La raison en est simple : le driver du clavier est entièrement réécrit en fond de mémoire. - Il a bien fallu trouver de la place pour implanter les nouvelles routines permanentes pour le curseur clignotant, les touches à répétitions et les abréviations.

Bref : le bilan final est tout à fait positif.

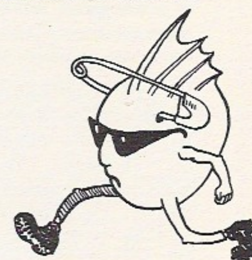
Créer un fichier de 1200 enregistrements de 120 octets... Plus de 80 Koctets de programme sur la disquette système... Trier 1000 fiches en 7 secondes... en une seule ligne BASIC... L'imprimante qui travaille (presque) toute seule...

Des rêves qui deviennent réalités !

Roger Brousmiche

mathématiques

De vrais petits monstres !

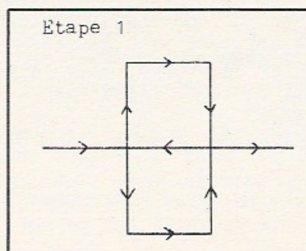


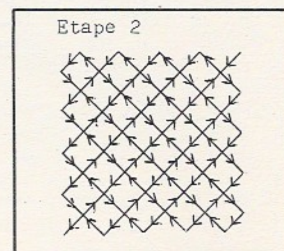
Les mathématiques, c'est bien connu, sont la science de la rigueur pure : des hypothèses, des raisonnements, des conclusions... que de l'abstrait (une droite n'a pas de de réalité physique, elle a une équation). Heureusement, au niveau historique, les mathématiques se sont rarement développées ainsi : il y a très souvent des choses bizarres, introduites par l'audace de certains (nombres dits complexes, calcul intégral...) qui ont contraint à réfléchir sur des notions délicates.

Enfin TRACE n'a pas pour but de former ses lecteurs à la philosophie des mathématiques ! Toujours est-il que, vers les années 1900, un certain nombre de "petits monstres" fut introduit. Une fois étudiés, ils sont allés rejoindre la galerie des curiosités ou amusements mathématiques, qu'on ouvre parfois pour ébahir le profane. L'étude de certains de ces petits monstres, appelés objets fractals, a connu ces derniers temps un regain d'intérêt sous l'impulsion d'un chercheur d'I.B.M. (tiens ! tiens ! ils ne se contentent pas de fabriquer des ordinateurs ceux-là).

Mais qu'est-ce au juste qu'un objet fractal ?
On trouve dans "Penser les mathématiques" (collection Points, Sciences, Seuil, page 228) la définition suivante "dont la forme est soit extrêmement irrégulière, soit extrêmement interrompue ou fragmentée, quelle que soit l'échelle d'examen". Tout s'éclaire donc !!! Revenons aux années 1900 pour voir définir deux objets dits fractals :

1) la courbe de Peano (1890)

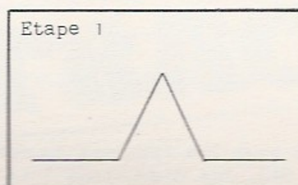


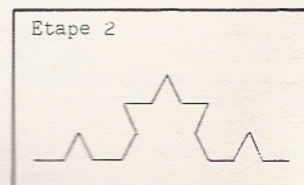


Etapes 3,4,.. on continue

Cette courbe - c'est-à-dire ce qu'on obtient "à la limite", en répétant indéfiniment le procédé - a la propriété bizarre de finir par remplir un carré. Eh oui : il y a des courbes tellement courbes qu'elles finissent par remplir des surfaces !! Je vous l'avais dit : un petit monstre !

2) la courbe de Von Koch (1904)

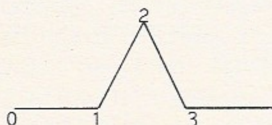




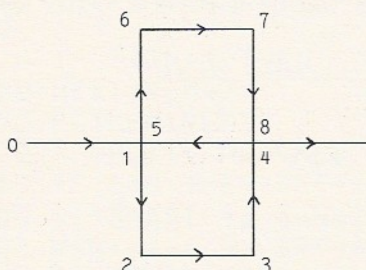
Etapes 3,4,... on continue

Au travers de ces exemples, on réalise mieux qu'un objet fractal est obtenu par la répétition systématique d'un procédé de construction. Essayez de faire tout seul l'étape n° 3 de la courbe de Peano... si vous y parvenez, cet article ne vous concerne pas. A la limite, revendez votre ordinateur, vous n'en avez pas besoin ! Sinon, chargez dans un premier temps le programme de tracé automatique de droite en langage machine (voir TRACE n° 5) puis tapez le programme ci-joint.

Il est important de comprendre la méthode utilisée dans ce programme pour "fractaliser" n'importe quoi. A priori, on pense à la récursivité, mais ce n'est pas le point fort du BASIC. Reprenons les exemples ci-dessus à l'étape n° 1 et numérotons les sommets 0,1,2,3 ... Repérons les angles et les longueurs de chaque segment du tracé, par rapport au segment joignant le point de départ et le point d'arrivée.

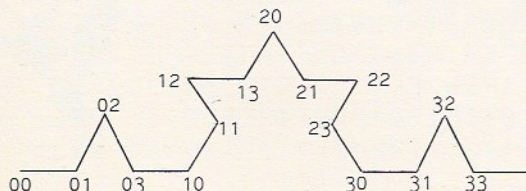


Origine du Segment	Angle	Longueur
0	0 (=A(Ø))	1/3 (=D(Ø))
1	60 (=A(1))	1/3 (=D(1))
2	-60 (=A(2))	1/3 (=D(2))
3	0 (=A(3))	1/3 (=D(3))



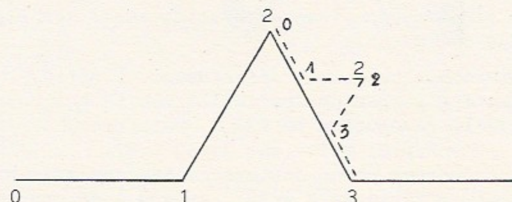
Origine du Segment	Angle	Longueur
0	0 A(Ø)	1/3 D(Ø)
1	-90° A(1)	1/3 D(1)
2	0 A(2)	1/3 D(2)
3	+90° A(3)	1/3 D(3)
4	-180° A(4)	1/3 D(4)
5	+90° A(5)	1/3 D(5)
6	0° A(6)	1/3 D(6)
7	-90° A(7)	1/3 D(7)
8	0° A(8)	1/3 D(8)

Ceci nous aide à comprendre ce qui se passe aux étapes suivantes de la fractalisation. Prenons l'étape 2 de la courbe de Von Koch. Il y a 17 points et 16 segments. Le dernier point n'a pas d'importance : seuls les 16 points de départ des segments seront importants. Numérotons ces 16 points en base 4 (= nombre de segments de l'étape 1), par un nombre de deux chiffres (les deux chiffres sont fondamentaux et deux numéro de l'étape).



Prenons un segment quelconque, celui qui commence par 21 par exemple (c'est 21-22)..Les chiffres 2 et 1 s'interprètent de la façon suivante :
2 signifie que le segment 21-22 de l'étape 2 provient d'un segment commençant par 2 à l'étape 1 (c'était 2-3).
1 signifie que le segment 21 de l'étape 2 (segment 21-22) correspond à un segment commençant par 1 de l'étape 1 (segment 1-2), sur le segment 2-3 de l'étape 1.

Le dessin suivant sera peut être plus clair :



Une autre façon d'exprimer cela est de dire que, par rapport au segment joignant le point de départ au point d'arrivée, l'angle que fait le segment d'origine 2-1 est :

A(2) + A(1)
→ Valeur de rotation à la 2ème étape
→ Valeur de rotation à la 1ère étape

Quant à la longueur de ce segment c'est D(2)*D(1) (pour les courbes de Peano et de Von Koch, ces longueurs sont les mêmes, quels que soient les segments considérés à une étape).

Prenez d'autres exemples, d'autres numéros d'étapes, vous constaterez que, pour "fractaliser" un motif formé, à l'étape n° 1 de N+1 points, il suffit de connaître deux tableaux :

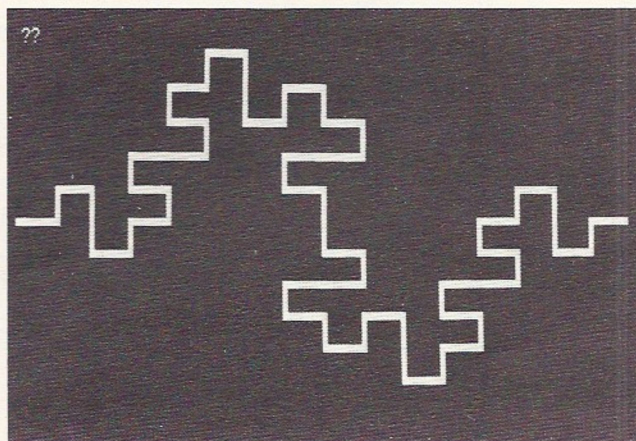
A(N) : angles des N segments
D(N) : longueurs des N segments

(par rapport au segment joignant les points de départ et d'arrivée, qui doivent être distincts). A la Kème étape, en numérotant les somments, sauf le dernier, par un nombre de K chiffres en base N, par rapport au segment joignant le point de départ au point d'arrivée,
on tourne de : A( ) + A( ) + ... + A( )
on a une longueur de : D( ) * D( ) * ... * D( )

chiffres de la décomposition en base N du numéro de l'origine du segment

Voilà vous savez tout sur la méthode utilisée, mais certaines précisions sur le fonctionnement du programme seront peut être les bienvenues.

Après initialisations (lignes 1070 à 1110) le programme principal appelle un menu (lignes 1180 à 1310).



Courbe de Sierpinski

Vous avez le choix entre :

- courbe de Von Koch
- courbe de Peano
- courbe de Sierpinski
- fractal défini par l'utilisateur.

Dans les trois premiers cas, les lignes 1320 à 1390 lisent le nombre de segments et les coordonnées-écran de leurs extrémités (data en lignes 2260-2290) nécessaires pour tracer le motif de départ qu'on va fractaliser.

Dans le dernier cas (fractal utilisateur), les lignes 1400-1760 demandent :

- le nombre N+1 de points définissant le motif,
- le nombre d'étapes de fractalisation (2 à 4 compte tenu de la résolution du TRS... mais, même sur un Apple, on ne peut aller que jusqu'à 5).
- les coordonnées écran des extrémités des segments du motif. x varie de Ø à 127, y de Ø à 44 car la dernière ligne d'écran est réservée au texte. Introduire une coordonnée négative permet de modifier les coordonnées précédentes (le programme demande donc de définir un N+1ième point, pour rectifier éventuellement le dernier point). Le motif est tracé au fur et à mesure qu'on introduit les points. Il ne s'agit que d'une version un peu modifiée d' un sous-programme utilisé pour déformer un motif en un autre (voir TRACE n° 5).

Arrivé à ce point, que le fractal soit pré-programmé ou fixé par l'utilisateur, le motif est défini par :

- le nombre N+1 de points du motif à l'étape 1.
- leurs coordonnées-écran (tableaux X(N),Y(N)).

Le programme convertit alors ces coordonnées en angles et longueurs (tableaux A et D, sous-programme 1870-1990 appelé en ligne 1150. La particularité de ce sous-programme est de calculer les angles et longueurs vrais alors que les x et y écran, sur TRS80, ne correspondent pas aux mêmes unités. D'où un coefficient 2 par lequel sont multipliés tous les y qui interviennent dans les calculs.

Cela fait, le programme effectue la fractalisation proprement dite (sous-programme 2000-2180 appelé en ligne 1160 et appelant lui-même le sous-programme de décomposition en base N, lignes 2190-2250). Le numéro de chaque étape s'inscrit en haut, à gauche de l'écran. Le tracé terminé, ce numéro est remplacé par ?? ce qui invite l'utilisateur à taper sur n'importe quelle touche pour passer à l'étape suivante de fractalisation.

Je vous donne en plus quelques exemples de fractals-utilisateur qui me semblent assez jolis :

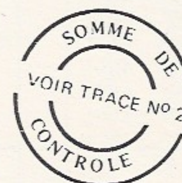
1/N=6 (0,44)(60,44)(30,18)(90,18)(60,44)(120,44)
2/N=7 (0,22)(63,22)(63,2)(63,22)(63,42)(63,22)(126,22)
3/N=6 (0,44)(42,44)(42,23)(84,23)(84,44)(126,44)
4/N=4 (60,25)(60,20)(70,20)(70,25)
5/N=6 (0,44)(63,26)(42,44)(84,44)(63,26)(126,44)
6/N=9 (0,23)(63,23)(42,5)(84,5)(63,23)(42,41)(84,41)(63,23)(126,23)

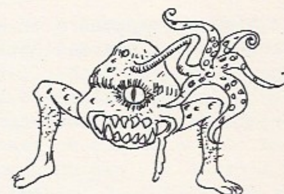
A vous maintenant de créer vos fractals... Ces "petits monstres" ne demandaient qu'une chose, se laisser apprivoiser !

La procédure pour exploiter le programme peut être résumée ainsi :

- Préciser le Memory Size selon votre configuration (voir TRACE numéro 5, page 28).
- Charger le programme SEGMENT.
- Modifier la ligne 210 selon la taille mémoire disponible.
- Faire RUN.
- Charger le programme FRACTAL.
- Faire RUN.

```
1000 '--------------------------------------------------------   15
1010 ' OBJETS FRACTALS - (C) Gerald Grandpierre et TRACE   !   0B
1020 '--------------------------------------------------------   01
1030 ' Configuration : modele 1 ou 3 - 16 a 48k de RAM      !   F6
1040 ' Cassette ou disquette                                 !   EC
1050 ' Utilise la routine SEGMENT parue dans TRACE no 5     !   E2
1060 '--------------------------------------------------------   D8
1070 ' INTIALISATIONS  CE
1080 CLS:REM40
1090 CLEAR 100 : DEFINT I-K,U-Y:REM94
1100 DIM X(20),Y(20),A(20),D(20),U(4):REM9A
1110 PI=3.14159 : U(0)=1:REMBD
1120 ' PROGRAMME PRINCIPAL  9C
1130 GOSUB 1180:REM37
1140 ON C GOSUB 1320,1320,1320,1400:REM78
1150 GOSUB 1870:REM1D
1160 GOSUB 2000:REM21
1170 FOR N=1 TO 200 : NEXT N : GOTO 1070:REMA8
1180 'MENU  60
1190 CLS:REMD2
1200 PRINT TAB(24) "OBJETS FRACTALS":REMF4
1210 PRINT TAB(24) STRING$(15,"*"):REM30
1220 PRINT : PRINT "VOUS POUVEZ CHOISIR ENTRE :":REM57
1230 PRINT : PRINT"  1)COURBE DE VON KOCH":REM91
1240 PRINT "  2) COURBE DE PEANO":REM17
1250 PRINT "  3) COURBE DE SIERPINSKI":REM7E
1260 PRINT "  4) FRACTAL DEFINI PAR L'UTILISATEUR":REM60
1270 PRINT : PRINT "VOTRE CHOIX (1 A 4) ?";:REMA8
1280 A$=INKEY$:REMF8
```

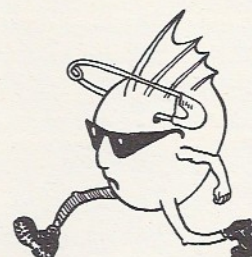






```
1290 IF A$<"1" OR A$>"4" THEN 1280:REM99
1300 C=VAL(A$) : NF=3 : CLS:REM90
1310 RETURN:REM4B
1320 ' FRACTALS PRE-PROGRAMMES  D3
1330 FOR I=1 TO C:REMF9
1340 READ N:REME6
1350 FOR J=0 TO N:REMDA
1360 READ X(J),Y(J):REM0D
1370 NEXT J:REMD0
1380 NEXT I:REMC7
1390 RETURN:REMFB
1400 ' MOTIF DEFINI PAR L'UTILISATEUR  83
1410 CLS:REMF5
1420 INPUT "NOMBRE DE POINTS (>2)";N:REM2F
1430 IF N<=2 THEN 1410 ELSE N=N-1:REMDC
1440 INPUT "NOMBRE D'ITERATIONS (2,3 OU 4)";NF:REMD5
1450 IF NF<2 OR NF>4 THEN 1440:REM24
1460 CLS:REMC3
1470 FOR I=0 TO N+1:REM65
1480 PRINT @960,"POINT" ;I+1; "ABSCISSE X ?";:REM88
1490 GOSUB 1770 : X(I)=VAL(B$):REM1C
1500 PRINT @960,STRING$(60,32);:REM1B
1510 PRINT @960,"POINT" ;I+1; "ORDONNEE Y ?";:REM5C
1520 GOSUB 1770 : Y(I)=VAL(B$):REMFD
1530 PRINT @960,STRING$(60,32);:REMFD
1540 IF X(I)<0 OR Y(I)<0 THEN GOSUB 1650 : GOTO 1480:REME8
1550 IF X(I)>127 OR Y(I)>44 THEN 1480:REM64
1560 IF I=N AND X(I)=X(0) AND Y(I)=Y(0) THEN I=I-1 : GOTO 1580:R
EM3C
1570 IF I<>N+1 THEN GOSUB 1600:REME8
1580 NEXT:REM47
1590 RETURN:REM32
1600 ' DROITE ALLUMEE  BA
1610 IF I=0 THEN SET(X(0),Y(0)) : RETURN:REM8A
1620 U(1)=X(I-1) : U(2)=Y(I-1) : U(3)=X(I) : U(4)=Y(I):REM7A
1630 U(0)=1 : W=USR(VARPTR(U(0))):REM61
1640 RETURN:REM00
1650 ' DROITE EFFACEE  88
1660 IF I=0 THEN RETURN:REM45
1670 IF I=1 THEN RESET(X(0),Y(0)) : I=I-1 : RETURN:REMAE
1680 U(1)=X(I-2) : U(2)=Y(I-2) : U(3)=X(I-1) : U(4)=Y(I-1):REM3E

1690 U(0)=0 : W=USR(VARPTR(U(0))):REM26
1700 I=I-1:REMF0
1710 IF I=1 THEN SET(X(0),Y(0)) : RETURN:REM25
1720 FOR J=0 TO I-2:REM6C
1730 U(1)=X(J) : U(2)=Y(J) : U(3)=X(J+1) : U(4)=Y(J+1):REM0A
1740 U(0)=1 : W=USR(VARPTR(U(0))):REMF3
1750 NEXT J:REM53
1760 RETURN:REM88
1770 ' REPONSES NUMERIQUES  10
1780 EC=980 : B$="":REM4F
1790 A$=INKEY$ : IF A$="" THEN 1790:REM17
1800 IF A$>="0" AND A$<="9" THEN 1860:REME8
1810 IF (A$="+" OR A$="-") AND B$="" THEN 1860:REMF6
1820 IF ASC(A$)=13 THEN RETURN:REM0D
1830 IF B$="" THEN 1790:REM2A
1840 IF ASC(A$)=8 THEN B$=LEFT$(B$,LEN(B$)-1) : PRINT @EC," "; :
 EC=EC-1:REM7B
1850 GOTO 1790:REM61
1860 B$=B$+A$ : EC=EC+1 : PRINT @EC,A$; : GOTO 1790:REMAD
```

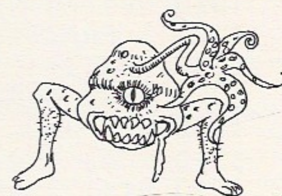








```
1870 ' ANGLES ET DISTANCES  AB
1880 DX=X(N)-X(O) : DY=Y(N)-Y(O) : D=SQR(DX*DX+4*DY*DY):REME5
1890 FOR I=0 TO N-1:REMBE
1900 IF DX=0 THEN A=-PI*2/SGN(DY) : GOTO 1920:REMED
1910 A=ATN(-2*DY/DX) : IF DX<0 THEN A=A+PI:REM6E
1920 FOR I=0 TO N-1:REMAD
1930 X=X(I+1)-X(I) : Y=Y(I+1)-Y(I):REM78
1940 D(I)=SQR(X*X+4*Y*Y)/D:REMAD
1950 SI=2*(Y*DX-X*DY) : CO=X*DX+4*Y*DY:REM49
1960 IF CO=0 THEN A(I)=-PI/2*SGN(SI) : GOTO 1980:REM1C
1970 A(I)=ATN(-SI/CO) : IF CO<0 THEN A(I)=A(I)+PI:REM7A
1980 NEXT I:REM6D
1990 RETURN:REMA1
2000 ' FRACTALISATION  29
2010 FOR K=1 TO NF:REMFC
2020 Z1=X(0) : Z2=Y(0) : U(1)=X(0) : U(2)=Y(0):REME7
2030 CLS:REM87
2040 PRINT @0,K;:REM2D
2050 FOR I=0 TO N[K-1:REM01
2060 R=I : A1=A : D1=D:REMF2
2070 GOSUB 2190:REM85
2080 Z3=Z1+D1*COS(A1) : U(3)=INT(Z3+.5):REM68
2090 Z4=Z2-.5*D1*SIN(A1) : U(4)=INT(Z4+.5):REM26
2100 IF U(1)<0 OR U(1)>127 OR U(2)<0 OR U(2)>47 OR U(3)<0 OR U(3
)>127 OR U(4)<0 OR U(4)>47 THEN 2120:REMAB
2110 U(0)=1 : W=USR(VARPTR(U(0))):REM7F
2120 U(1)=U(3) : U(2)=U(4) : Z1=Z3 : Z2=Z4:REM1A
2130 NEXT I:REMD6
2140 PRINT @0,"??";:REM51
2150 A$=INKEY$:IF A$="" THEN 2150:REMB6
2160 NEXT K:REMB6
2170 PRINT @0,"FIN";:REMD4
2180 RETURN:REME2
2190 ' DECOMPOSITION EN BASE N  6A
2200 FOR J=1 TO K:REM87
2210 Q=INT(R/N) : R1=R-Q*N:REM77
2220 A1=A1+A(R1) : D1=D1*D(R1):REMD1
2230 R=Q:REMCA
2240 NEXT:REMB1
2250 RETURN:REM9C
2260 ' DATA COURBES  24
2270 DATA 4, 0, 10, 42, 10, 63, 28, 84, 10, 126, 10:REMB4
2280 DATA 9, 20, 0, 50, 15, 20, 30, 50, 45, 80, 30, 50, 15, 80,
0, 110, 15, 80, 30, 110, 45:REM03
2290 DATA 8, 0, 23, 31, 23, 31, 7, 63, 7, 63, 23, 63, 39, 95, 39
, 95, 23, 127, 23:REM46
```

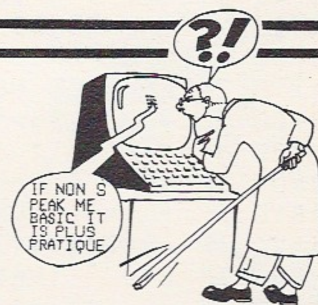




Gérald Grandpierre

enseignement

Parler-vous un bon franssais?



Un test effectué par ordinateur provoque toujours chez les élèves une motivation certaine et crée une stimulation que seul l'attrait de la machine peut provoquer. Ce test peut paraître simpliste, mais il a été mis au point pour des élèves qui arrivent dans des classes professionnelles avec des prérequits plutôt réduits. De par sa conception et sa présentation, l'élève peut recevoir directement son résultat et voir

sur l'écran les erreurs commises. Quant au professeur, il dispose en fin de test d'un synoptique général que seul l'ordinateur peut donner et qu'en aucun cas il n'obtiendrait avec les méthodes de corrections traditionnelles.

A PROPOS DU PROGRAMME

L'entrée des données se fait par la fonction INKEY$ avec possibilité de corrections par retour arrière.

Décomposition du programme :

- Lignes 10 - 140 : Initialisation. Entrée des nom et prénom de l'élève. Présentation.
- Lignes 150 - 340 : Déroulement général du test.
- Lignes 350 - 430 : Tableau des résultats.
- Lignes 440 - 520 : Tableau général en fin de test d'une classe.
- Lignes 530 - 620 : Entrées des réponses aux questions.
- Lignes 630 - 710 : Entrées des données d'ordre général.
- Lignes 720 - 790 : Appréciation des réponses.
- Lignes 800 - 810 : Temporisation.
- Lignes 820 - 1040 : Phrases et solutions.
- Lignes 1050 - 1110 : Impression des résultats généraux.
- Lignes 1120 - 1280 : Sous-programme de présentation.

Choix des verbes a mettre a la 3 eme personne de l'indicatif

Enoncer - Prevoir - Relater - Emettre - Formuler - Exposer
Raconter - Notifier - Divulguer - Annoncer - Confier - Dicter
Transmettre - Communiquer - Rapporter - Reveler - Colporter
Proferer - Avouer - Ordonner - Prononcer - Predire - Declarer

Bien des gens DISENT des betises. Le bon eleve sait de

source sure que l'on raconte.... une histoire

Bonne reponse ! ! !

Nota : Ce programme a été prévu pour un TRS80 doté des caractères minuscules. (Les utilisateurs ne possédant pas cette option devront entrer les caractères minuscules en mode SHIFT pour conserver une somme de contrôle correcte avec le programme de contrôle VERCS. Ndlr)

Les lecteurs n'ayant pas d'imprimante pourront supprimer les lignes 520, 530, 1050 et 1110 et remplacer le contenu de la ligne 510 par END.




```
1 '--------------------------------------------  FF
2 '      Test de francais                     !  FE
3 '  (C) 1983 Roger Henrotay et TRACE         !  FD
4 '--------------------------------------------  FC
5 ' Configuration : modele 3 ou modele 1 !  FB
6 ' avec minuscules. 16 a 48k de RAM.         !  FA
7 ' Cassette ou disquette.                    !  F9
8 ' Imprimante en option.                     !  F8
9 '--------------------------------------------  F7
10 CLS:CLEAR 1000:DIM NO$(25),PR$(25),E$(25),F$(25):REM3D
20 GOSUB 1120:CLS:REMD9
30 PRINT @ 134,"Choisissez dans la liste des verbes celui qui":R
EM0C
40 PRINT @ 326,"convient le mieux et mettez-le a la 3 eme person
ne de l'":REMAE
50 PRINT @ 596,"INDICATIF  PRESENT":REM1C
60 PRINT @ 910,"'ESPACE' quand vous etes pret : ";:REM02
70 R$="":R$=INKEY$:IF R$="" THEN 70:REM54
80 IF R$<>CHR$(32) THEN 70:REM23
90 CLS:FOR K=0 TO 23:E$(K)="":F$(K)="":NEXT:REM83
100 PRINT @ 266,"";:PRINT "Quel est votre nom : ";:REM5F
110 GOSUB 630:NO$(M)=R$:REM84
120 IF NO$(M)="fin" OR NO$(M)="FIN" THEN 440:REMBA
130 PRINT @ 464,"et votre prenom : ";:REM6F
140 GOSUB 630:PR$(M)=R$:REM61
150 REM  A4
160 CLS:PRINT "Choix des verbes a mettre a la 3 eme personne de
l'indicatif":REMF3
170 FOR X=0 TO 127:SET(X,6):SET(X,22):NEXT:REMA2
180 FOR Y=6 TO 22:SET(0,Y):SET(127,Y):NEXT:REM95
190 PRINT @ 193,"Enoncer - Prevoir - Relater - Emettre - Formule
r - Exposer";:REM9F
200 PRINT @ 257,"Raconter - Notifier - Divulguer - Annoncer - Co
nfier - Dicter";:REM92
```

```
210 PRINT @ 321,"Transmettre - Communiquer - Rapporter - Reveler
 - Colporter";:REMCB
220 PRINT @ 385,"Proferer - Avouer - Ordonner - Prononcer - Pred
ire - Declarer";:REM65
230 PRINT @ 580,"Bien des gens DISENT des betises. Le bon eleve
sait de":REM19
240 PRINT @ 708,"source sure que l'on";:REMC9
250 FOR K=1 TO 23:REM12
260 READ A$,B1$:REM48
270 C$=LEFT$(B1$,4):REM75
280 D$=RIGHT$(B1$,2):REM6B
290 GOSUB 540:REMB3
300 GOSUB 720:REMA9
310 PRINT @ 911,STRING$(35,32):REM02
320 PRINT @ 730,STRING$(40,32):REMFD
330 NEXT:REM2E
340 R(M)=INT((R(M)/23)*100):REME7
350 CLS:PRINT"Tableau des resultats":PRINT STRING$(35,45):REM0E
360 PRINTTAB(10)"Reponse exacte";TAB(30)"Votre reponse":REM32
370 FOR K=0 TO I:REMB6
380 PRINT TAB(10)E$(K);TAB(30)F$(K):REMFD
390 NEXT:REMF2
400 PRINT"Votre resultat est donc : ";R(M)"/100":REM9C
410 PRINT"Voulez-vous un autre test : ";:REMB7
420 GOSUB 630:REM31
430 IF R$="o" OR R$="O" THEN M=M+1:RESTORE:GOTO 90:REME2
440 REM:Resultats generaux sur ecran  81
450 CLS:REMB9
460 PRINT"Tableau des resultats";TAB(29)"General";TAB(38)"Vocabu
laire";TAB(51)"Conjugaison":REMD4
470 PRINT STRING$(63,45):REM64
480 FOR K=0 TO M:REM44
490 PRINT NO$(K);TAB(15)PR$(K);TAB(32)R(K);TAB(42)S(K);TAB(57)T(
K):REMBA
500 NEXT:REM84
510 PRINT:PRINT "'ESPACE' si vous voulez imprimer":REM6D
520 GOSUB 630:REMCC
530 IF R$="" THEN 1050 ELSE END:REMEF
540 J=1:PRINT @ 729,A$:REM33
550 B2$="":B2$=LEFT$(A$,1):REM75
560 R$="":R$=INKEY$:IF R$="" THEN 560:REM34
570 IF R$=CHR$(32) THEN RETURN:REME1
580 B2$=B2$+R$:REM72
590 R=LEN(B2$):REMAD
600 PRINT @ 729+J,R$;:REM1E
610 IF R$=CHR$(8) THEN B2$=LEFT$(B2$,R-2):J=J-2:REM09
620 J=J+1:GOTO 560:REMC9
630 REM:Entrees de donnees  C2
640 X$="":R$="":REM20
650 X$="":X$=INKEY$:IF X$="" THEN 650:REMC8
660 IF X$=CHR$(32) THEN RETURN:REM81
670 R$=R$+X$:REM56
680 R=LEN(R$):REM75
690 IF X$=CHR$(8) THEN R$=LEFT$(R$,R-2):REM9A
700 PRINT X$;:REMD9
710 GOTO 650:REM10
720 REM:Calculs des bonnes et mauvaises reponses  68
730 IF B2$=B1$ THEN 780:REM28
740 IF C$<>LEFT$(B2$,4) THEN PRINT @ 911,"Erreur, consultez un d
ictionnaire";:S(M)=S(M)+1:GOTO 770:REM7F
750 IF D$<>RIGHT$(B2$,2) THEN PRINT @ 911,"Veuillez revoir vos c
onjugaisons";:T(M)=T(M)+1:GOTO 770:REM10
```





```
760 IF LEN(B$)<>LEN(B2$) THEN PRINT @ 911,"Revoyez votre orthogr
aphe";:GOTO 770:REM19
770 E$(I)=B1$:F$(I)=B2$:I=I+1:GOTO 800:REME3
780 R(M)=R(M)+1:REM3E
790 PRINT @ 911,"Bonne reponse ! ! !";:REM5E
800 FOR T=1 TO 500:NEXT:REMEF
810 RETURN:REM41
820 DATA "e.......... une opinion","emet":REM6D
830 DATA "e.......... un plan d'action","expose":REM27
840 DATA "e.......... une regle","enonce":REM79
850 DATA "r.......... une histoire","raconte":REM96
860 DATA "r.......... une aventure vecue","relate":REME0
870 DATA "f.......... un souhait","formule":REM4F
880 DATA "n.......... une decision","notifie":REM93
890 DATA "a.......... une nouvelle","annonce":REM86
900 DATA "c.......... un avis","communique":REM35
910 DATA "d.......... un secret","divulgue":REM35
920 DATA "t.......... un ordre","transmet":REM7C
930 DATA "r.......... des critiques","rapporte":REM5F
940 DATA "r.......... un complot","revele":REM73
950 DATA "c.......... des potins","colporte":REM9B
960 DATA "p.......... des menaces","profere":REMBA
970 DATA "p.......... un discours","prononce":REM08
980 DATA "d......... le courrier a une secretaire","dicte":REMD6

990 DATA "a.......... ses fautes","avoue":REMC3
1000 DATA "o.......... une attaque militaire","ordonne":REMAC
1010 DATA "p.......... le proche avenir","predit":REM34
1020 DATA "p......... le temps des prochains jours","prevoit":REM
AC
1030 DATA "d.......... la guerre ou son amour","declare":REM0B
1040 DATA "c......... ses peines a un ami","confie":REMF1
1050 REM:Resultats generaux sur imprimante  1C
1060 LPRINT"Tableau des resultats";TAB(35)"General";TAB(48)"Voca
bulaire";TAB(65)"Conjuguaison":REM79
1070 LPRINT STRING$(80,45):REM0D
1080 FOR K=0 TO M:REME9
1090 LPRINT NO$(K);TAB(20)PR$(K);TAB(37)R(K);TAB(52)S(K);TAB(70)
T(K):REM65
1100 NEXT:REM29
1110 END:REM26
1120 REM:Presentation  D6
1130 FOR X=1 TO 126:SET(X,0):SET(X,7):NEXT:REM11
1140 FOR Y=0 TO 7:SET(0,Y):SET(126,Y):NEXT:REM05
1150 PRINT @ 72,"Test de vocabulaire et conjuguaison de verbes";
:REM44
1160 PRINT @ 256,"Ce test permet de jauger vos connaissances":RE
M79
1170 PRINT @ 404,"du point de vue des conjuguaisons":REM90
1180 PRINT @ 532,"et de l'usage du verbe idoine":REMB5
1190 PRINT @ 640,"Pendant l'execution du test il est ";:REM34
1200 PRINT @ 936,"d'appuyer sur 'ENTER'";:REM56
1210 PRINT @ 788,STRING$(22,32);:REM38
1220 FOR T=1 TO 100:NEXT:REM4E
1230 PRINT @ 788,"FORMELLEMENT interdit";:REMFD
1240 FOR T=1 TO 100:NEXT:REM3A
1250 PRINT @ 960,"'ESPACE' pour continuer";:REM42
1260 A$="":A$=INKEY$:REM55
1270 IF A$="" THEN 1210:REM6B
1280 RETURN:REM69
```



Roger Henrotay



Des rubans sans fin (ou presque)

Vous avez peut-être gardé les vieilles cassettes-rubans de votre imprimante, cassettes donnant une impression si pâle qu'elles donnaient... une pâle impression de votre travail et que bon gré mal gré vous avez dû remplacer.

Le petit truc ci-après permet de recharger en encre les cassettes épuisées et leur redonner une vigueur toute printanière en les régénérant et ce sinon éternellement tout au moins un nombre appréciable de fois d'où une économie non négligeable.

La description qui suit a été testée sur les rubans-cassettes d'une imprimante "PRINTER VI". Mais il est certain à quelques détails près, que la même opération peut être envisagée pour les rubans de tout autre imprimante utilisant des rubans encreurs en cassettes ou même en bobines.
Le matériel utilisé est simple et se trouve à peu près partout.

Le principal est l'ENCRE qui doit impérativement être d'un type adapté : pour ma part, j'ai utilisé avec succès de l'encre à tampons sans huile, coloris noir, et de marque "PELIKAN", la référence portée sur la boîte carton d'emballage étant 4K. J'avoue n'avoir pas essayé un produit similaire d'une autre marque, celle-ci ayant donné toute satisfaction aux premiers essais. Je me suis donc bien gardé d'en changer, en adoptant le principe... : Le MIEUX est (souvent) l' ennemi du BIEN.

Le flacon encreur "Pélikan" a aussi l'avantage de comporter un bouchon applicateur de la largeur du ruban ou presque, ce qui le rend directement utilisable, sans avoir recours à l'utilisation d'un feutre intermédiaire, ce qui simplifie les choses.

Pour réencrer, il faut faire défiler manuellement le ruban à l'aide du bouton moleté situé sur le côté droit de la cassette, en le tournant dans le sens des aiguilles d'une montre et en le maintenant devant l' embout du flacon placé en position verticale tout en pressant légèrement ce dernier. Je dis bien "légèrement" sous peine de se retrouver avec de beaux pâtés sur la table de travail et ce qui est pire, avec de belles taches sur le bas de vos pantalons qu'un "omo" puissant (sinon sapiens) aura bien du mal à faire disparaître, l'encre en question ayant par nature une humeur de type particulièrement et résolument indélébile.

Donc après avoir saisi la cassette d'une main et placé le petit bouton moleté entre le pouce et l'index de l'autre, vous chercherez désespérément une troisième main pour tenir et presser (délicatement) le flacon pour en faire sortir l'encre. Vous avez, bien sûr, la solution d'appeler un voisin complaisant, une épouse dévouée, ou votre secrétaire de service (on peut sans inconvénient intervertir les facteurs qualificatifs !). D'autres solutions sont possibles, sans déranger tous ces éventuels et dévoués collaborateurs.

Pour ma part, voici celle que j'ai adoptée :
La cassette est serrée verticalement dans un petit étau dont les mors sont garnis de deux morceaux de feutre pour la protéger (ce n'est pas indispensable, mais prudent). Cela suffit, me direz-vous, car on a trouvé la main manquante de tout à l'heure. Oui mais on se lasse très vite de tourner entre les doigts nus le petit bouton moleté surtout qu'il faut faire 380 tours complets pour que le ruban défile entièrement (dans le cas de la cassette printer VI). Alors il faut confectionner une manivelle.

Pour ma part, j'utilise une manivelle très courte de jouet "MECCANO". Sur l'extrémité de l'axe est serré un manchon d'accouplement d'axes (toujours Meccano) dont le trou d'un côté a été agrandi au diamètre du bouton moleté de la cassette, soit, toujours pour la Printer VI, à 6 mm.

Voilà, il ne reste plus qu'à insérer le bouton moleté, en forçant légèrement, dans la manivelle et à faire défiler le ruban en tournant, tout en maintenant en contact le bouchon encreur du flacon et le ruban. Il faut environ 380 tours soit deux à trois minutes et le ruban est réencré.

Deux remarques importantes :
1) Le ruban a tendance lorsqu'il est garni d'encre fraîche à patiner : dans ce cas, il faut insérer dans le petit espace situé à droite du bouton moleté, entre le corps de la cassette et la pièce mobile un petit coin en bois ou en carton. Pour ma part, j'utilise un petit morceau d'allumette géante coupé en biseau au couteau.

2) une fois le ruban réencré, il est recommandé de faire à nouveau 280 tours de manivelle mais cette fois-ci sans encrer, ceci au bout de quelques heures pour mieux répartir l'encre dans la totalité du ruban. En effet, le ruban est plié en accordéon dans la cassette et une attente même de quelque jours après l'encrage et avant l'utilisation est souhaitable. Dans ce but, j'utilise plusieurs rubans que je garde en réserve après les avoir rechargés.

On peut apporter beaucoup de perfectionnements à ce petit truc : par exemple, on peut envisager de faire un montage complet en fixant le flacon d'encre... et même motoriser le tout...
Ah... j'allais oublier de vous dire... La première fois que j'ai utilisé le système, mes doigts ressemblaient à ceux d'un écolier de 1925 utilisant un encrier à godet et une plume "Sergent Major" (les jeunes, ceux plus près de 77 ans, me comprendront... et ceux, plus près de 7 ans, demanderont éventuellement aux autres de leur expliquer) ; alors il faudra dans ce cas, bien improbable car je suis sûr que vous êtes plus doués que moi, faire comme l'illustre Pilate et utiliser la pierre du même nom pour retrouver des "paluches" à peu près présentables. On peut aussi adopter la crème Ajax récurrente qui m'a donné de bons résultats. (Pub non payée !!)

Les fabricants de rubans et autres vendeurs de cassettes vont m'en vouloir et j'ai un peu honte à l'avouer, mais une cassette ainsi traitée permet le même temps d'utilisation à chaque recharge qu'une cassette neuve. A ce jour, une de mes cassette a déjà été rechargée 11 fois sans donner aucun signe de défaillance du ruban ce qui est un hommage rendu à leur solidité. Un flacon d'encre (de 28 ml) permet environ 10 recharges (y compris l'encre perdue la première fois sur les doigts et ailleurs. Il coûte une vingtaine de francs. Quant à l'Ajax éventuel, vous avez toujours la ressource de le prendre en douce à la cuisine familiale.
Il n'y a finalement que de petites économies.

Jacques Dannenmuller

utilitaire

Gardez trace de l'usage de vos variables (II)

Vous avez lu TRACE n° 5, et le programme générateur de références croisées (pour le modèle 2) vous a certainement intrigué, puis interessé,... mais ne disposant que d'un modeste modèle 1 ou 3, vous avez à regret tourné la page.

Repartons donc à zéro.

Les références d'un programme BASIC sont les variables simples, les tableaux et les branchements (goto, gosub, etc.), lus dans le listing. Attention : les variables de plus de 2 caractères seront scindées. Ex : NCLE% donnera NC et LE%.

Pour être utilisables, les références doivent être affichées sur l'écran, ou mieux sur l'imprimante, sous la forme d'un tableau trié, avec en regard de chaque item la liste, également triée des numéros des lignes concernées.

Les références sont dites croisées lorsque les branchements sont précisés par les numéros de lignes 'source' et 'destination'. Si, par exemple, une sous-routine est implantée à la ligne 1000, la référence GOSUB 1000 sera suivie des numéros des lignes contenant ce branchement.

Le descripteur des références est un prodigieux outil pour le "décryptage" d'un programme inconnu. Il est indispensable au programmeur, pour le remodelage d'un programme existant (même s'il en est l'auteur !).

Comment utiliser le programme ? Quel est le matériel requis ?

Tout modèle 1 ou 3, niveau 2, 16 K ou plus convient. La disquette est souhaitée mais non indispensable. Par contre l'imprimante est pratiquement nécessaire, sinon il faut se résoudre à recopier le texte affiché à l'écran...

Pour commencer, il faut charger le programme à analyser, puis chaîner ou fusionner le descripteur des références que nous appelerons CREF/TXT.

Les utilisateurs de disquettes ont déjà compris qu'il suffit de faire LOAD pour le programme à analyser, puis MERGE 'CREF/TXT' qui aura été sauvé en ASCII.

Les adeptes de la cassette trouveront les instructions nécessaires dans TRACE n° 4, à la page 21 pour le modèle 1 et la page 24 pour le modèle 3 (fusion de programmes BASIC).

La première ligne de CREF/TXT porte le numéro 64000, afin d'éviter les collisions avec les lignes du programme à analyser.

Cela fait, vous tapez RUN 64000 et le message suivant apparaît à l'écran :

```
* 2272 REFERENCES maximum
(e=ELSE, g=GOTO, r=RESUME, s=GOSUB, t=THEN)
DEPUIS LA LETTRE (A MAJUSCULE), JUSQU'A (z minuscule)
? A,z
FAUT-IL IMPRIMER (O/N) ? O
NOM DU PROGRAMME ? -
```

```
A    1900 1910 2060
A$   1280 1290 1300 1790 1800 1810 1320 1840 1860
 2150
A(   1100 1960 1970 2220
A1   2060 2080 2090 2220
B$   1490 1520 1780 1810 1830 1840 1860
C    1140 1300 1330
CO   1950 1960 1970
D    1880 1940 2060
D(   1100 1940 2220
D1   2060 2080 2090 2220
DX   1880 1900 1910 1950
DY   1880 1900 1910 1950
EC   1780 1840 1860
I    1090 1330 1380 1470 1480 1490 1510 1520 1540
 1550 1560 1570 1610 1620 1660 1670 1680 1700
```

Le nombre maximum des références dépend de la taille mémoire et de la longueur du programme.

La ligne suivante décrit les abréviations mnémoniques (lettres minuscules) des branchements.

A la troisième ligne on vous demande d'entrer deux lettres. Si vous répondez A,z toutes les références seront recherchées et éditées (Un simple <ENTER> donne le même résultat).

Pour A,Z les branchements seront ignorés.
Pour a,z seuls les branchements seront édités.
Pour J,J seules les références des variables dont le nom commence par J sont reconnues, etc.

En général, il suffit de taper <ENTER> et tout se passe bien si la taille mémoire est suffisante. Sinon vous serez stoppé par un BS ERROR en cours de lecture, et vous relancerez le programme pour réduire le nombre de lettres à référencer.

A titre d'exemple nous avons pu analyser les plus longs programmes de TRACE n° 4 (7 à 8 K) en deux passes maximum : A,Z puis a,z (sous niveau 2, 16 K !).

Sous disk-basic 48K, l'utilitaire de Tandy TEST2/BAS, de 14500 octets et 1280 références a été décodé en une seule passe.

En prime, la longueur du programme est écrite au bas du tableau.

En fin de programme, vous devez répondre à la question

UN NOUVEAU RUN 64000 (O/N) ?

Si vous répondez N=non, CREF/TXT est effacé, et seul subsiste en mémoire le programme utilisé.

Il faut aussi signaler quelques rares anomalies :
1) La variable AS n'est pas reconnue (BASIC disque)
2) La ligne Ø n'est jamais mentionnée parmi les numéros de lignes.

3) Les espaces ne sont pas admis entre le nom d'un tableau et la parenthèse ouverte qui suit. Par exemple, DIM US (5) sera repercuté par US au lieu de US(
4) Une instruction du type X!=2E6 répercute X! et E6 De même DD#=3D17 donne DD# et D1.

CREF/TXT est construit autour de trois modules superposés :

D'abord la lecture du programme, avec la recherche des variables et des branchements, et leur mise en fiche dans F$() en même temps que le numéro de ligne dans F(). La vitesse de lecture et de traitement est de 10 à 12 octets par seconde.

Ensuite vient le tri vertical et alphanumérique de toutes les références. La durée du tri est évidemment très variable... Il s'agit d'un tri BASIC dit rapide (hum !) ; vous pouvez avantageusement le remplacer par une routine binaire (voir TRACE n° 3, page 52).

Enfin les références sont affichées à l'écran et éventuellement par l'imprimante, après un tri numérique (tri horizontal) des numéros de lignes concernées.
Ce tri est éliminatoire, c'est-à-dire que si une variable est redéfinie plusieurs fois dans une même ligne, ce numéro de ligne ne sera écrit qu'une seule fois.

Le format d'édition sur l'imprimante est prévu pour 80 caractères maximum par ligne. Pour changer le format il suffit de modifier le contenu de la variable LL=80 à la ligne 64060.

Il est bien évident que vous tenterez d'abord de référencer CREF/TXT, et cela ne marchera pas parce que la lecture s'arrête à la ligne 64000. Ce point d'arrêt est défini à la ligne 64060 par XF=64000.

Vous procéderez comme suit :
Après le lancement du programme, à la question :
    A PARTIR DE LA LETTRE... ?
vous répondrez <BREAK> puis <CLEAR> et vous taperez la sentence :
    XF=70000 : CONT   <ENTER>

Le reste n'est plus qu'un jeu d'enfant...

N'oubliez cependant pas de faire un CSAVE ou un SAVE, avec l'option ASCII, avant le premier RUN !

La liste du programme CREF contient des textes en caractères minuscules. Les lecteurs ne possédant pas cette fonctionnalité devront frapper les textes correspondants en appuyant sur la touche SHIFT s'ils veulent conserver le "checksum" correct (Ndlr).

```
64000 '  06
64010 ' CREF/TXT - (C) Roger Brousmiche et TRACE  FC
64011 ' Configuration de base : modele 1 avec minuscules  FB
64012 '--------------------------------------------------  FA

64020 '  F2
64030 CLEAR 50 : CLEAR MEM/3 : DEFINT I-L : I=MEM*.11:REM59
64040 CR$="( e=ELSE, g=GOTO, r=RESUME, s=GOSUB, t=THEN )":REMD3
64050 L1=0 : L2=0 : L3=0 : I1=0 : DIM F$(I),F(I),F1(I/5),I(1,20)
:REM83
64060 XF=64000 : R=65536 : S=32768 : LL=80:REM78
64070 CLS : PRINT "*"I"REFERENCES maximum." : PRINT CR$:REMA2
64080 PL$="A" : DL$="z" : INPUT"DEPUIS LA LETTRE <A MAJUSCULE>,
JUSQU'A <z minuscule> ";PL$,DL$:REMB9
64090 INPUT "FAUT-IL IMPRIMER (O/N) ";P$ : IF P$<>"O" GOTO 64120
:REMB8
64100 INPUT"NOM DU PROGRAMME ";N$:REMFB
64110 '****** LECTURE  98
64120 PRINT@ 320,"JE LIS :"; : A=PEEK(16548)+PEEK(16549)*256:REM
A5
64130 D=A+1 : GOSUB 64810 : B=PEEK(A)+J*256 : IF C>=XF OR B=0 TH
EN 64350:REMDA
64140 D=A+2 : GOSUB 64810 : C=J : D=A+3 : GOSUB 64810 : C=C+J*25
6:REM3D
64150 IF B-A>260 THEN B=B-R:REME1
64160 FOR D=A+4 TO B-2 : B$="" : GOSUB 64810:REM04
64170 IF J=34 GOSUB 64600 : GOTO 64330:REMEF
64180 IF J=147 OR J=136 : D=B-2 : GOTO 64330:REM20
64190 IF J=149 : B$="e" : GOTO 64630:REM14
64200 IF J=141 : B$="g" : GOTO 64630:REM10
64210 IF J=145 : B$="s" : GOTO 64630:REMF6
64220 IF J=202 : B$="t" : GOTO 64630:REMF1
64230 IF J=159 : B$="r" : GOTO 64630:REMDE
64240 IF J<33 OR J>90 THEN 64330:REMDD
64250 IF J>64 AND J<91 AND LEN(B$)<2 GOSUB 64800 : GOTO 64250:RE
MBA
64260 IF B$="AS" : D=D-1 : GOTO 64330:REM02
64270 IF LEN(B$)=0 THEN 64330:REMEF
64280 IF J>47 AND J<58 AND LEN(B$)=1 GOSUB 64800:REM61
```



```
64290 IF J=40 OR J=37 OR J=33 OR J=35 OR J=36 GOSUB 64800:REMD5
64300 IF J=40 AND RIGHT$(B$,1)<>"(" GOSUB 64800:REM3F
64310 A$=LEFT$(B$,1) : IF A$<PL$ OR A$>DL$ THEN D=D-1 : GOTO 64330:REM18
64320 K=K+1 : F(K)=C : F$(K)=B$ : PRINT @ 413,C;B$;K;CHR$(30) : D=D-1:REMD7
64330 NEXT D : GOSUB 64840 : GOTO 64130:REM5A
64340 '****** CALCUL LONGUEUR PROGRAMME  B1
64350 E=D : IF E<0 : E=E+R:REM7D
64360 E=E-PEEK(16548)-PEEK(16549)*256+1 : IF B : E=E-8:REM11
64370 '****** TRI VERTICAL  93
64380 IG=1 : L=K : B$="" : PRINT @ 512,"JE TRIE :";:REM28
64390 IF L<=IG THEN 64470:REM9A
64400 B$=F$((IG+L)/2) : I=IG : J=L:REMF6
64410 IF F$(I)<B$ : I=I+1 : GOTO 64410:REM37
64420 IF F$(J)>B$ : J=J-1 : GOTO 64420:REM2A
64430 IF I>J THEN 64450:REM94
64440 GOSUB 64490 : D=F(I) : F(I)=F(J) : F(J)=D : I=I+1 : J=J-1 : IF I<=J THEN 64410 ELSE PRINT @ 607,F$(I);CHR$(30):REMF9
64450 IF I<L : IP=IP+1 : I(0,IP)=I : I(1,IP)=L:REM38
64460 L=J : GOTO 64390:REM01
64470 IF IP=0 THEN 64530:REM36
64480 IG=I(0,IP) : L=I(1,IP) : IP=IP-1 : GOTO 64390:REM39
64490 L1=VARPTR(F$(I)) : L2=VARPTR(F$(J)):REM11
64500 FOR L3=0 TO 2 : I1=PEEK(L1+L3) : POKE L1+L3,PEEK(L2+L3):REMB5
64510 POKE L2+L3,I1 : NEXT L3 : RETURN:REMDA
64520 '****** APPEL DES AFFICHAGES  FC
64530 IG=K : I=0 : F$="######" : IF P$="O" GOSUB 64670:REMEE
64540 K=1 : I=I+1 : IF I>IG GOSUB 64700 : GOTO 64890:REM1E
64550 PRINT F$(I);TAB(6); : GOSUB 64680 : F1(K)=F(I):REMC2
64560 IF F$(I+1)=F$(I) : I=I+1 : K=K+1 : F1(K)=F(I) : GOTO 64560:REM5A
64570 GOSUB 64740 : PRINT : IF P$="O" LPRINT:REMF4
64580 GOTO 64540:REM30
64590 '****** TRAITEMENT GUILLEMETS  B6
64600 D=D+1 : GOSUB 64810 : IF (J<>34 AND D<=B-2) THEN 64600:REM41
64610 RETURN:REM10
64620 '****** TRAITEMENT GOSUB, GOTO, RESUME, THEN, ELSE  98
64630 D=D+1 : GOSUB 64810:REM65
64640 IF J=32 THEN D=D+1 : GOSUB 64810 : GOTO 64640 ELSE IF (J>47 AND J<58) GOSUB 64800 : GOTO 64640:REMBE
64650 IF LEN(B$)=1 THEN B$="" : GOTO 64170 ELSE 64310:REM5C
64660 '****** IMPRIMANTE  70
64670 LPRINT "REFERENCES de "N$ : LPRINT CR$ : LPRINT" " : RETURN:REMBB
64680 IF P$="O" LPRINT USING"%    %";F$(I); : K1=6:REMBA
64690 RETURN:REMC0
64700 PRINT N$ " occupe" E "octets en memoire vive.":REM9D
64710 IF P$="O" LPRINT " " : LPRINT N$ " occupe" E "octets en memoire vive." : LPRINT " ":REMD1
64720 RETURN:REMA2
64730 '****** TRI HORIZONTAL ET AFFICHAGE  2A
64740 A=-1 : FOR J=0 TO K-1 : L=J : FOR L1=J+1 TO K : IF F1(L1)=F1(L) : F1(L1)=A:REM6C
64750 IF F1(L1)<F1(L) : L=L1:REM80
64760 NEXT L1 : D=F1(L) : F1(L)=F1(J) : F1(J)=D : NEXT J:REMFC
64770 FOR J=1 TO K : IF F1(J)<>A PRINT F1(J); : IF P$="O" LPRINT TAB(K1) USING F$;F1(J); : GOSUB 64860:REM2A
64780 NEXT J : RETURN:REM5A
```



```
64790 '****** TRAITEMENT REFERENCES ET AJUST. ENTIER  ED
64800 B$=B$+CHR$(J) : D=D+1:REM4E
64810 IF D<S THEN J=PEEK(D) ELSE J=PEEK(D-R):REMF2
64820 RETURN:REM3D
64830 '****** AJUST. ENTIER ET TAB( ) IMPRIMANTE  C5
64840 IF B<S THEN A=B ELSE A=B-R:REM58
64850 RETURN:REM1F
64860 K1=K1+6:IF K1>LL-6 LPRINT:K1=6:REMB2
64870 RETURN:REM0B
64880 '****** FIN DU PROGRAMME  93
64890 A$="" : INPUT"UN NOUVEAU RUN 64000 (O/N) ";A$:REM92
64900 IF A$="O" THEN RUN 64000 ELSE IF A$="N" CLEAR 50 : DELETE
64000-64910 : END:REM14
64910 GOTO 64890:REMDD
```



Roger Brousmiche

utilitaire

Mini-Zap deviendra grand

Qui n'a pas entendu parler de SUPERZAP ? Il s'agit d'un programme utilitaire livré avec le système d'exploitation NEWDOS qui, entre autres choses, permet de "voir" les informations telles qu'elles sont sur la disquette. Le système standard TRSDOS livré avec le modèle 3 ne permettant pas une telle possibilité, voici une petit programme qui remplit deux fonctions : lire et afficher le contenu d'un secteur sur l'écran et éventuellement le recopier sur la seconde unité (naturellement, nous attendons avec impatience qu'un lecteur nous propose une extension permettant l'édition du secteur lu ! Ndlr).

MINIZAP est composé de deux programmes : l'un est en BASIC et l'autre en assembleur. Pour cette dernière raison, il est conseillé de posséder l'éditeur-assembleur EDTASM (le lecteur intéressé par la lecture des secteurs d'une disquette se doit de l'avoir dans sa panoplie...).
Le programme objet doit obligatoirement être sauvé sous le nom "MINIZAP/CMD". Cela étant fait, il suffit de lancer le programme BASIC sans se soucier du MEMORY SIZE (effectué automatiquement). Les adresses d'implantation sont prévues pour une configuration de 48 K octets de mémoire vive, ce qui est implicite avec des disquettes.

Une fois le programme lancé, un message s'affiche :

DRIVE, PISTE, SECTEUR ?

Si le numéro de piste introduit est égal à zéro (la première piste portant le numéro 1), alors le numéro de secteur est relatif au début de la disquette. Dans le cas contraire (no piste différent de 0), le programme calcule lui-même l'adresse du secteur (piste * 18 + secteur) et en affiche le contenu sur l'écran avec les mentions : SDP (secteur dans la piste), PST (piste), DRV (drive) et SDD (secteur dans la disquette). Les commandes autorisées sont les suivantes :

<;> : affichage du secteur suivant
<-> : affichage du secteur précédent
<X> : retour à "DRIVE, PISTE, SECTEUR ?"
<E> : le secteur lu sur le drive 0 est écrit sur le drive 1 à la même position
<S> : sortie et retour au BASIC

MINIZAP utilise les adresses des routines d'accès disque décrites dans la documentation du manuel Tandy. La routine DECOD a été prise dans le livre "La pratique du TRS80", volume 2. Le programme assembleur est abondamment commenté afin de permettre à ceux qui le désirent de faire mieux, et pourquoi pas, de refaire SUPERZAP sous TRSDOS !

```
1 '-----------------------------------  FF
2 '          M I N I Z A P           !  FE
3 '                                  !  FD
4 ' Configuration : modele 3         !  FC
5 '                 48 K, DISQUE     !  FB
6 '-----------------------------------  FA
7 ' (C) 1983 Louis Martin & TRACE    !  F9
8 '-----------------------------------  F8
10 CLEAR 1000:REM7D
20 POKE 16561,&H60:REM38
30 POKE 16562,&HEA:REM0D
```



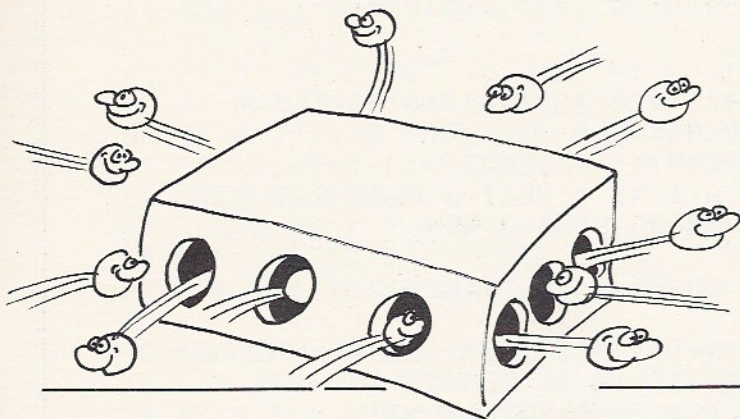
```
40 CMD"L","MINIZAP/CMD":REM38
50 DEFUSR1=&HEA60: A=USR1(0):REM3A
60 CLS:REM40
70 INPUT "DRIVE,PISTE,SECTEUR";D,P,S:REM01
80 GOSUB 300 : GOSUB 330:REM2B
90 P=P+INT(S/18):REMAF
100 S=S-INT(S/18)*18:REM66
110 GOSUB 300 : GOSUB 330:REM0D
120 POKE &HE706,D : POKE &HE710,P : POKE &HE711,6:REMCC
130 GOSUB 350:REM55
140 DEFUSR1=&HEA6C: A=USR1(0):REMCD
150 PRINT "SDP"; : PRINT @64,S;:REM02
160 PRINT; : PRINT @192,"PST"; : PRINT @256,P;:REM89
170 PRINT; : PRINT @384,"DRV"; : PRINT @448,D;:REM90
180 PRINT; : PRINT @576,"SDD"; : PRINT @640,S4;:REM57
190 A$=INKEY$ : IF A$=";" THEN 250:REM5C
200 IF A$="-" THEN 260:REM9C
210 IF A$="E" THEN 270:REM79
220 IF A$="S" THEN 280:REM60
230 IF A$="X" THEN RUN 60:REMF7
240 GOTO 190:REME9
250 S4=S4+1 : GOSUB 290 : GOTO 80:REM90
260 S4=S4-1 : GOSUB 290 : S=S-2 : GOTO 80:REMCF
270 POKE &HE706,1 : GOSUB 290 : S=S-1 : GOSUB 350 : GOSUB 300 :
DEFUSR1=&HEAF6: A=USR1(0) : GOTO 190:REM73
280 DEFUSR1=&HEABF: A=USR1(0) : END:REM77
290 S=PEEK(&HE70A) : RETURN:REM58
300 IF S4<0 THEN P=39 : S=17:REM92
310 IF S4>719 THEN S=0 : P=0:REM8D
320 RETURN:REM2D
330 S4=P*18+S:REMB1
340 RETURN:REM19
350 POKE &HE70A,S:REM16
360 RETURN:REM05
```




```
E700            00100           ORG     0E700H  ;
E700 4D         00110 TITRE     DEFM    'MINIZAP/CMD'
E701 49
E702 4E
E703 49
E704 5A
E705 41
E706 50
E707 2F
E708 43
E709 4D
E70A 44
E70B 03         00120           DEFB    03H     ;
4424            00130 OPEN      EQU     4424H   ;
05D9            00140 INPUT     EQU     05D9H   ;
4409            00150 ERROR     EQU     4409H   ;
4436            00160 READ      EQU     4436H   ;
4439            00170 WRITE     EQU     4439H   ;
4428            00180 CLOSE     EQU     4428H   ;
EA60            00190           ORG     0EA60H  ;
EA60 CDE4EA     00200 DEBUT     CALL    INIT    ;
EA63 0600       00210           LD      B,0     ; LRL=256 octets
EA65 CD2444     00220           CALL    OPEN    ; ouverture
EA68 C2EBEA     00230           JP      NZ,FAUTE ; si l'ouverture se passe mal
EA6B C9         00240           RET             ;
EA6C 3EFF       00250           LD      A,0FFH  ;
EA6E 320CE7     00260           LD      (0E70CH),A
EA71 CDE4EA     00270           CALL    INIT    ;
EA74 CD3644     00280           CALL    READ    ; 1 secteur disque --> buffer
EA77 2072       00290           JR      NZ,FAUTE ; si lecture se passe mal
EA79 00         00300           NOP             ;
EA7A CDC901     00310           CALL    1C9H    ; effacement de l'ecran
EA7D 3E00       00320           LD      A,0     ; 1ere ligne ecran
```

```
EA7F 0610      00330          LD     B,10H    ; compteur boucle BCL0
EA81 11053C    00340          LD     DE,3C05H ; pointeur ecran
EA84 2133E7    00350          LD     HL,0E733H ; pointeur buffer du FCB
EA87 C5        00360 BCL0     PUSH   BC       ; sauvegarde compteur B
EA88 CDC8EA    00370          CALL   AFFICH   ; affichage no de ligne
EA8B C610      00380          ADD    A,10H    ; no de ligne suivante
EA8D F5        00390          PUSH   AF       ; sauvegarde registre A
EA8E 13        00400          INC    DE       ; avance pour un blanc
EA8F 0608      00410          LD     B,8      ; compteur BCL1
EA91 7E        00420 BCL1     LD     A,(HL)   ; 1 car du buffer dans A
EA92 CDC8EA    00430          CALL   AFFICH   ; et affichage
EA95 23        00440          INC    HL       ; adresse car suivant
EA96 7E        00450          LD     A,(HL)   ; car suivant dans A
EA97 CDC8EA    00460          CALL   AFFICH   ; et affichage
EA9A 23        00470          INC    HL       ; pour prochain passage
EA9B 13        00480          INC    DE       ; avance un blanc
EA9C 10F3      00490          DJNZ   BCL1     ; boucle pour 8*2 car
EA9E 01F0FF    00500          LD     BC,-10H  ; retour arriere de 16 car
EAA1 09        00510          ADD    HL,BC    ;
EAA2 0610      00520          LD     B,10H    ; compteur BCL2
EAA4 7E        00530 BCL2     LD     A,(HL)   ; 1 car dans A
EAA5 FE21      00540          CP     21H      ; voir si car est imprimable
EAA7 3804      00550          JR     C,POINT  ; sinon, le remplacer
EAA9 FE80      00560          CP     80H      ; par un point
EAAB 3802      00570          JR     C,CONT   ; a l'affichage
EAAD 3E2E      00580 POINT    LD     A,2EH    ;
EAAF 12        00590 CONT     LD     (DE),A   ; affichage du car
EAB0 13        00600          INC    DE       ; position suivante ligne
EAB1 23        00610          INC    HL       ; position suivante buffer
EAB2 10F0      00620          DJNZ   BCL2     ; suite
EAB4 010500    00630          LD     BC,5     ; ajoute 5 a DE pour
EAB7 EB        00640          EX     DE,HL    ; passer a la ligne suiv.
EAB8 09        00650          ADD    HL,BC    ;
EAB9 EB        00660          EX     DE,HL    ;
EABA F1        00670          POP    AF       ; no de ligne dans A
EABB C1        00680          POP    BC       ; compteur dans B
EABC 10C9      00690          DJNZ   BCL0     ; suite boucle BCL0
EABE C9        00700          RET             ;
EABF CDE4EA    00710          CALL   INIT     ;
EAC2 CD2844    00720          CALL   CLOSE    ; fermeture FCB
EAC5 2024      00730          JR     NZ,FAUTE ; si fermeture se passe mal
EAC7 C9        00740          RET             ;
EAC8 F5        00750 AFFICH   PUSH   AF       ;
EAC9 F5        00760          PUSH   AF       ;
EACA E6F0      00770          AND    0F0H     ; effacement quartet faible
EACC 07        00780          RLCA            ; quartet fort prend la place
EACD 07        00790          RLCA            ; du quartet faible
EACE 07        00800          RLCA            ;
EACF 07        00810          RLCA            ;
EAD0 CDDBEA    00820          CALL   DECOD    ;
EAD3 F1        00830          POP    AF       ; A reprend sa valeur d'entree
EAD4 E60F      00840          AND    0FH      ; effacement quartet fort
EAD6 CDDBEA    00850          CALL   DECOD    ;
EAD9 F1        00860          POP    AF       ; A reprend sa valeur d'entree
EADA C9        00870          RET             ;
EADB C690      00880 DECOD    ADD    A,90H    ; en sortie, A contient le
EADD 27        00890          DAA             ; code ASCII du car qu'il
EADE CE40      00900          ADC    A,40H    ; contenait en entree
EAE0 27        00910          DAA             ;
EAE1 12        00920          LD     (DE),A   ; affichage du car dont le
EAE2 13        00930          INC    DE       ; code ASCII vient d'etre
EAE3 C9        00940          RET             ; calcule et avance 1 position
EAE4 1100E7    00950 INIT     LD     DE,0E700H
EAE7 2133E7    00960          LD     HL,0E733H
EAEA C9        00970          RET             ;
EAEB F680      00980 FAUTE    OR     80H      ; affichage code erreur par
EAED CD0944    00990          CALL   ERROR    ; la routine DOS
EAF0 CD2844    01000          CALL   CLOSE    ; fermeture du FCB
EAF3 C3191A    01010          JP     1A19H    ; retour au BASIC
EAF6 1100E7    01020          LD     DE,0E700H
EAF9 CD3944    01030          CALL   WRITE    ; ecriture d'un secteur disque
EAFC 20ED      01040          JR     NZ,FAUTE ; si ecriture se passe mal
EAFE C9        01050          RET             ;
EA60           01060          END    DEBUT
00000 TOTAL ERRORS
```

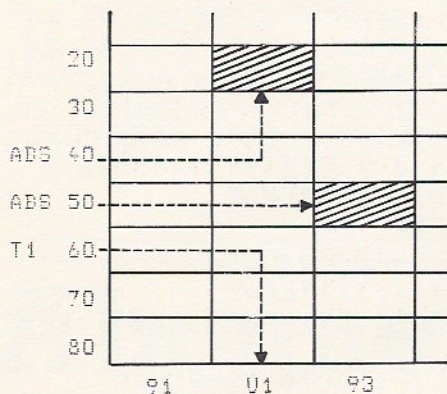
Louis Martin



La réflexion mise en boîte

Ce programme, inspiré d'un jeu appelé "black box", ne fait pas appel aux réflexes comme on pourrait le penser mais à la réflexion et aux réflexions ! Il se joue sur une grille de 8 par 8 dans laquelle sont dissimulés des obstacles. Selon la difficulté recherchée, vous pouvez choisir leur nombre entre 3 et 9. Ils ne peuvent être situés sur les bords de la grille que si ce nombre est supérieur à 5.

Le but du jeu est évidemment de retrouver ces obstacles et pour cela, vous avez deux moyens : utiliser la commande R pour donner le numéro de la case contenant l'obstacle suspecté (mais encore faut-il en être sûr !), ou bien tirer un projectile au début d'une ligne ou d'une colonne par la commande T. Ce projectile ne détruit pas l'obstacle mais, selon son comportement, il peut vous préciser où se trouve celui-ci. En effet, lorsqu'un obstacle est placé sur la ligne ou la colonne de tir, le projectile est tout simplement absorbé (noté ABS sur l'écran). Si par contre, l'obstacle est situé sur une ligne ou une colonne voisine, le projectile est réfléchi (dévié à 90 degrés). S'il rencontre un autre obstacle,



il sera absorbé ou dévié de nouveau... et il se peut ainsi qu'il revienne à son point de départ ! Avec un peu de réflexion... on peut déterminer à coup sûr l'emplacement obstrué.

```
ESSAI
BLANC          1   2   3   4   5   6   7   8
( B )  ABS 10                                  19
ESSAI      20                                  29
REEL   T 1 30                                  39
( R )  ABS 40                                  49
 TIR   ABS 50                                  T 3
( T )  ABS 60                                  T 4
           70                                  79
       T 2 80                                  89

( 9 OBSTACLES )  U 2  92  93  U 1  95  96  U 4  U 3

ESSAI NUMERO 4  REUSSI
TOTAL DES POINTS  :    100    ( +25 )
```

Sur l'exemple ci-dessus, le premier tir (ligne 50) est absorbé par l'obstacle situé en 53. Le second tir (ligne 40) est réfléchi par l'obstacle 53 puis absorbé par 22. Dans les deux cas, l'affichage ABS clignote (ces tirs ne coûtent pas de points). Enfin, le troisième tir (colonne 60) marqué T1, se réfléchit puis sort en 92. L'affichage clignote : T1 à l'entrée, U1 à la sortie.
Pour "essayer" une position (53 dans notre exemple), on utilisera la commande R. Vous disposez de 8 tirs "gratuits" au-delà desquels vous perdez 20 points par tir. Chaque obstacle découvert rapporte 25 points mais chaque essai manqué en coûte 10. Si vous trouvez tous les obstacles en utilisant moins de 8 projectiles, chacun de ceux qui restent donne un bonus de 15 points.
La commande B permet de matérialiser ou d'effacer une marque placée à une position donnée de la grille, pour repérer un point suspect, par exemple (ne coûte pas de points). La commande S permet de supprimer un obstacle déjà découvert mais devenu gênant. Elle ne coûte pas de points mais ne peut être utilisée qu'une seule fois. Enfin, la commande A que vous n'utiliserez probablement jamais (n'est-ce pas ?), permet d'abandonner le jeu et de découvrir les obstacles cachés.

A titre indicatif, le meilleur score réalisé par l'auteur est de 260 points (9 obstacles découverts, 3 projectiles inutilisés et 1 essai manqué).
Mais avant d'y réfléchir, absorbez-vous d'abord à entrer ce petit programme (non, il n'est pas long !).

```
1 ' Configuration d'essai : modele 1, niveau 2, 16k ** FF
2 ' (C) 1982 TRACE et Roger Sawicki **  FE
3 RANDOM : CLS : DEFINT A-Y : DEFSTR Z : DIM A(99):REM8A
4 INPUT "COMBIEN D'OBSTACLES ( 3  A  9 ) ";N:REMAD
5 IF N>9 OR N<3 THEN 4:REME9
6 IF N>5 THEN R=1:REMF2
7 CLS : FOR I=1 TO 8 : FOR J=1 TO 8 : A(10*I+J)=1:REM14
8 NEXT J,I : E=N : FOR J=1 TO N:REMFA
9 U=RND(6+2*R)+1-R : V=RND(6+2*R)+1-R:REM22
10 I=10*U+V : IF A(I)=1 THEN A(I)=9 : NEXT J ELSE 9:REMC0
11 FOR I=36 TO 116 : FOR J=6 TO 30 STEP 3:REM69
12 SET(I,J) : NEXT J,I:REME1
13 FOR I=36 TO 116 STEP 10 : FOR J=7 TO 29:REM30
14 SET(I,J) : NEXT J,I:REMDF
15 FOR I=1 TO 8 : J=64*I : K=5*I : PRINT @77+J,2*K;:REM46
16 PRINT @123+J,2*K+9; : PRINT @78+K,I;:REM35
17 PRINT @718+K,90+I; : NEXT I : Z="ESSAI":REM33
18 PRINT @0,Z; : PRINT @64,"BLANC"; : PRINT @128,"( B )";:REM65
19 PRINT @192,Z; : PRINT @256,"REEL"; : PRINT @320,"( R )";:REMF
3
20 PRINT @385,"TIR"; : PRINT @448,"( T )";:REMA4
21 PRINT @961,"TOTAL DES POINTS  :  ";:REM37
22 PRINT @704,"(";N;" OBSTACLES )";:REMB9
23 IF T<8 THEN 26:REMC6
24 PRINT @384,"VOUS"; : PRINT @448,"N'AVEZ"; : PRINT @512,"PLUS"
;:REM8A
25 PRINT @576,"DE"; : PRINT @640,"PROJECTILE";:REMBD
26 Z=INKEY$ : IF Z="" THEN 26:REM80
27 IF Z="A" THEN 71:REM70
28 FOR I=1 TO 60 : PRINT @896+I," "; : NEXT I:REM53
29 IF Z="B" THEN 34:REM6E
30 IF Z="R" THEN 39:REM58
31 IF Z<>"T" THEN 92:REM81
32 IF T>7 THEN TT=TT-20 : PRINT @984,TT;"   ( -20 )";:REM94
33 GOTO 52:REMEB
34 PRINT @833,"POSITION A VISUALISER ( + ) OU EFFACER ( - ) ";:R
EM5C
35 INPUT I : IF A(ABS(I))=0 THEN 75:REM02
36 GOSUB 89 : IF I>0 THEN 38 ELSE RESET(X+3,Y):REM10
37 RESET(X+4,Y+1) : RESET(X+5,Y) : GOTO 75:REM0D
38 SET(X+3,Y) : SET(X+4,Y+1) : SET(X+5,Y) : GOTO 75:REM1F
39 PRINT @833,"QUELLE POSITION VOULEZ-VOUS ESSAYER  ";:REME6
40 INPUT I : IF A(I)=0 THEN 75:REM27
41 F=F+1 : IF F<=N THEN 43:REM3F
42 TT=TT-10 : PRINT @984,TT;"   ( -10 )";:REM44
43 GOSUB 89 : IF A(I)=9 THEN 47:REMEC
44 FOR J=1 TO 4 : SET(X+4,Y) : FOR I=1 TO 199 : NEXT I:REMB2
45 RESET(X+4,Y) : FOR I=1 TO 99 : NEXT I,J : SET(X,Y):REM7E
46 GOSUB 91 : PRINT @913," MANQUE"; : GOTO 75:REM69
47 FOR J=1 TO N : IF D(J)=I THEN DD=1:REMD7
48 NEXT J : IF DD=1 THEN DD=0 : F=F-1 : GOTO 75:REM84
49 E=E-1 : D(E)=I : FOR I=1 TO 7 : SET(X+I,Y):REM40
50 SET(X+I,Y+1) : NEXT I : GOSUB 91 : PRINT @913,"REUSSI";:REM7B

51 TT=TT+25 : PRINT @984,TT;"   ( +25 )"; : GOTO 75:REMFF
52 PRINT @833,"POSITION DE DEPART "; : INPUT I:REMC7
53 IF I<1 OR I>98 THEN 75:REM55
54 IF A(I)<>0 THEN 75 ELSE K=I : GOSUB 85 : X=P:REM29
55 IF I<9 THEN M=10 : GOTO 58:REM61
56 IF I>90 THEN M=-10 : GOTO 58:REM64
57 IF 10*INT(I/10)=I THEN M=1 ELSE M=-1:REM3A
```

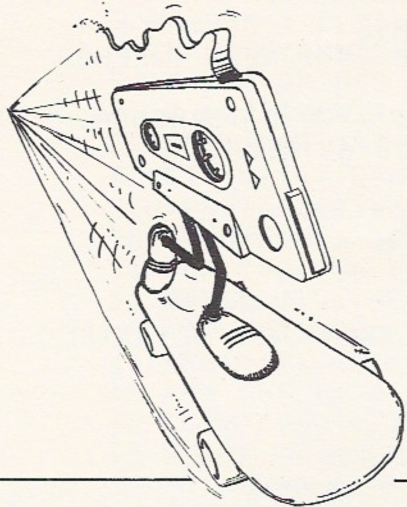


```
58 I=I+M : IF A(I)=0 THEN 67:REM65
59 IF A(I)=9 THEN 61:REM1C
60 IF A(I+M)<>9 THEN J=11*SGN(M) : GOTO 63:REM9E
61 FOR J=1 TO 3 : PRINT @X,"   "; : FOR I=1 TO 99 : NEXT I:REM90

62 PRINT @X,"ABS"; : FOR I=1 TO 199 : NEXT I,J : GOTO 75:REMDA
63 IF A(I+J)=9 AND A(I+2*M-J)=9 THEN I=K : GOTO 67:REMDD
64 IF A(I+J)=9 THEN M=M-J : GOTO 58:REMAC
65 IF A(I+2*M-J)=9 THEN M=J-M:REMC3
66 GOTO 58:REMC4
67 GOSUB 85 : Y=P : T=T+1 : Z=STR$(T):REM50
68 FOR J=1 TO 3 : PRINT @X,"    "; : PRINT @Y,"    ";:REM9C
69 FOR I=1 TO 99 : NEXT I : PRINT @X,"T"+Z; : PRINT @Y,"U"+Z;:RE
M5C
70 FOR I=1 TO 125 : NEXT I,J : GOTO 75:REME2
71 FOR I=11 TO 88 : IF A(I)=9 THEN 73:REMA5
72 NEXT I : PP=1 : GOTO 75:REMD5
73 FOR J=1 TO N : IF D(J)<>I THEN NEXT J ELSE 72:REM6D
74 GOSUB 89 : PRINT @(INT(Y/3))*64+INT(X/2)+2,"*"; : GOTO 72:REM
25
75 FOR I=1 TO 63 : PRINT @832+I," ";:REM35
76 NEXT I : IF PP=0 THEN 78:REM3D
77 PRINT @961,"VOUS ABANDONNEZ  :  0                    "; : GOTO 83
:REMAD
78 IF E<>0 THEN 23:REMD5
79 FOR I=1 TO 63 : PRINT @896+I," "; : NEXT I:REM1D
80 IF T>7 THEN 82 ELSE U=120-15*T : TT=TT+U:REM2A
81 PRINT @984,TT;"            "; : PRINT @832,"BONUS  :  ";U;:REMC
5
82 PRINT @896,"VOUS AVEZ TROUVE LES ";N;" OBSTACLES";:REM76
83 Z=INKEY$ : IF Z="" THEN 83:REM44
84 RUN:REM1E
85 IF I<9 THEN P=14+5*I : RETURN:REM8A
86 IF I>90 THEN P=268+5*I : RETURN:REM20
87 IF 10*INT(I/10)=I THEN P=73+6.4*I : RETURN:REM87
88 P=65.5+6.4*I : RETURN:REM6C
89 B=ABS(I) : V=B/10 : X=27+10*B-100*V:REMB2
90 Y=4+3*V : RETURN:REM53
91 PRINT @897,"ESSAI NUMERO";F; : RETURN:REMC8
92 IF Z<>"S" THEN 26:REM48
93 IF SR=1 PRINT @833,"PROCEDURE DEJA UTILISEE  "; : GOTO 26:REM
DE
94 PRINT @833,"OBSTACLE A SUPPRIMER "; : INPUT I:REM06
95 SR=1 : IF A(I)<>9 THEN 26:REM3D
96 GOSUB 89 : IF NO POINT(X+1,Y+1) THEN 26:REM16
97 PRINT @64*(INT(Y/3))+INT(X/2)+2,"S";:REM51
98 A(I)=1 : GOTO 75:REM8A
```



Roger Sawicki



Des cassettes avec un grand V

Le but de cet utilitaire est de recopier vos programmes BASIC de telle façon que vous puissiez les relire quatre fois plus rapidement (à 2000 BAUDS) sans aucune modification du matériel. La relecture à cette vitesse se fait automatiquement par la commande SYSTEM sans avoir à charger un programme au préalable... et vous bénéficiez par la même occasion du contrôle de "checksum" fourni gratuitement par le format "system"!

MODE D'EMPLOI

POUR L'ECRITURE EN 2000 BAUDS;
- Charger CRB (Recopie de Programme BASIC) et lancez-le.
- Charger ou frapper le programme BASIC à recopier.
- Activer CRB par : SYSTEM/31812 (version 16K).
- Donner le nom du programme (6 caractères obligatoires, complété éventuellement par des espaces).
- Préciser le nombre de copies à effectuer.
- Positionner le magnétophone en écriture et frapper sur une touche quelconque.

POUR LA RELECTURE D'UN PROGRAMME BASIC ECRIT PAR CRB.
- Baisser légèrement le volume du magnétophone.
- Frapper : SYSTEM *? <le nom du programme>
  (à chaque bloc lu correspond l'affichage d'un "#" sur l'écran).

PRINCIPE

CRB enregistre tout d'abord sur la cassette un "préchargeur" au format SYSTEM habituel (durée : 5 secondes), puis les pointeurs BASIC de début et de fin de programme, et enfin le programme BASIC lui-même. A la relecture, le préchargeur au format SYSTEM est lu et se lance automatiquement. C'est lui qui prend ensuite la direction des opérations et qui charge le programme dans le "buffer" BASIC. L'écriture et la lecture du programme principal utilisent une routine spéciale permettant de gérer la cassette à la vitesse de 2000 bauds.

Ce programme a fonctionné sans aucun problème sur mon TRS modèle 1 pendant plusieurs mois. J'ai même constaté une fiabilité accrue. Un ami l'utilise également sur un Vidéo Génie et en est très satisfait. Quant à moi, je l'ai "laissé tomber"... m'étant enfin offert un lecteur de disquettes ! Mais il faut bien penser aux autres !

Voici la liste de ce programme en BASIC (binaire sous forme de lignes DATA) ainsi que le "source" assembleur pour ceux qui voudraient en comprendre le fonctionnement.

```
1 '-----------------------------------------------------  FF
2 '      CRB (Chargement Rapide Basic)                  !  FE
3 '      (C) 1983 - Michel Fourcade et TRACE            !  FD
4 '      Configuration : modele 1, 16K, cassette        !  FC
5 '-----------------------------------------------------  FB
10 POKE 16561,68 : POKE 16562,124 : CLEAR 50:REM9F
20 PRINT "  LE PROGRAMME EN LANGAGE MACHINE QUI SE CHARGE (15 S)
 EST AUTO PROTEGE PAR UN MEMORY SIZE ET PERMET DE COPIER SUR CAS
SETTE LES PROGRAMMES EN BASIC A UNE VITESSE D'ENVIRON 2000 BAUDS
.":REM20
30 PRINT "  CHARGEZ OU TAPEZ VOTRE PROGRAMME BASIC, ESSAYEZ LE S
I VOUS LE DESIREZ, PUIS ACTIVEZ LE COPIEUR RAPIDE PAR:    SYSTE
M (ENTER) /31812 (ENTER) ET SUIVEZ LES INSTRUCTIONS.":REMBF
40 PRINT " POUR RELIRE UN PROGRAMME BASIC ENREGISTRE PAR LE COPI
EUR RAPIDEBAISSEZ LE NIVEAU DU MAGNETOPHONE ENTRE 4 ET 4.5 PUIS
TAPEZ:    SYSTEM (ENTER) NOM DU PROGRAMME (ENTER)":REMD9
50 PRINT "  PENDANT LE CHARGEMENT UN CONTROLE DE  'CHECKSUM' S'E
FFECTUERA,MODIFIEZ LE NIVEAU DE LECTURE EN CAS D'ERREUR. LE NOMB
RE DE ####VOUS PERMETTRA DE CONNAITRE LE NOMBRE DE BLOCS LUS.":R
EM7F
```

```
60 PRINT "  UNE FOIS LE PROGRAMME CHARGE VOUS POUVEZ LE LISTER,
L'EXECUTERPAR 'RUN' OU LE SAUVER PAR 'CSAVE' HABITUELS.":REM8F
70 PRINT "PATIENTEZ 15 SECONDES...   ";:REM91
80 FOR I=31812 TO 32767 : READ D% : POKE I,D% : NEXT I:REM4B
90 PRINT "===> "; : END:REMC6
100 DATA 42, 164, 64, 34, 149, 125, 42, 249, 64, 34:REMAA
110 DATA 151, 125, 205, 201, 1, 33, 163, 124, 205, 167:REM28
120 DATA 40, 17, 153, 125, 6, 6, 33, 44, 61, 213:REM40
130 DATA 205, 73, 0, 254, 32, 250, 98, 124, 254, 91:REM95
140 DATA 242, 98, 124, 209, 18, 119, 19, 35, 16, 235:REM4E
150 DATA 33, 53, 125, 205, 167, 40, 33, 149, 61, 54:REM83
160 DATA 32, 205, 73, 0, 254, 49, 250, 124, 124, 254:REM4E
170 DATA 58, 242, 124, 124, 33, 149, 61, 119, 230, 15:REM0E
180 DATA 50, 159, 125, 33, 74, 125, 205, 167, 40, 205:REM03
190 DATA 73, 0, 195, 160, 125, 69, 78, 82, 69, 71:REMA7
200 DATA 73, 83, 84, 82, 69, 77, 69, 78, 84, 32:REMED
210 DATA 65, 32, 50, 48, 48, 48, 32, 66, 65, 85:REMF8
220 DATA 68, 83, 32, 68, 69, 83, 32, 80, 82, 79:REME3
230 DATA 71, 82, 65, 77, 77, 69, 83, 32, 69, 78:REMD0
240 DATA 32, 66, 65, 83, 73, 67, 13, 86, 69, 82:REMD3
250 DATA 83, 73, 79, 78, 32, 50, 46, 48, 32, 32:REMD4
260 DATA 68, 85, 32, 49, 53, 47, 49, 48, 47, 56:REMB8
270 DATA 50, 32, 32, 80, 65, 82, 32, 32, 77, 73:REMCF
280 DATA 67, 72, 69, 76, 32, 32, 70, 79, 85, 82:REMA9
290 DATA 67, 65, 68, 69, 13, 13, 13, 68, 79, 78:REM9B
300 DATA 78, 69, 90, 32, 76, 69, 32, 78, 79, 77:REM85
310 DATA 32, 68, 85, 32, 80, 82, 79, 71, 82, 65:REM91
320 DATA 77, 77, 69, 32, 40, 54, 32, 67, 65, 82:REM87
330 DATA 65, 67, 84, 69, 82, 69, 83, 41, 58, 32:REM73
340 DATA 0, 13, 13, 78, 79, 77, 66, 82, 69, 32:REMA8
350 DATA 68, 69, 32, 67, 79, 80, 73, 69, 83, 58:REM55
360 DATA 32, 0, 13, 77, 69, 84, 84, 69, 90, 32:REM98
370 DATA 76, 69, 32, 77, 65, 71, 78, 69, 84, 79:REM3D
380 DATA 80, 72, 79, 78, 69, 32, 69, 78, 32, 77:REM3A
390 DATA 79, 68, 69, 32, 69, 78, 82, 69, 71, 73:REM2A
400 DATA 83, 84, 82, 69, 77, 69, 78, 84, 13, 69:REM20
410 DATA 84, 32, 65, 80, 80, 85, 89, 69, 90, 32:REM2A
420 DATA 83, 85, 82, 32, 85, 78, 69, 32, 84, 79:REM14
430 DATA 85, 67, 72, 69, 32, 46, 0, 0, 0, 0:REMFC
440 DATA 0, 32, 32, 32, 32, 32, 32, 0, 205, 132:REM48
450 DATA 2, 62, 85, 205, 100, 2, 6, 6, 33, 153:REM5D
460 DATA 125, 126, 35, 205, 100, 2, 16, 249, 33, 18:REM57
470 DATA 127, 126, 254, 60, 194, 234, 125, 205, 100, 2:REMB8
480 DATA 35, 126, 71, 205, 100, 2, 175, 79, 62, 2:REMA1
490 DATA 95, 35, 126, 87, 129, 79, 122, 205, 100, 2:REM28
500 DATA 29, 32, 244, 35, 126, 87, 129, 79, 122, 205:REME5
510 DATA 100, 2, 16, 245, 121, 205, 100, 2, 35, 195:REM33
520 DATA 183, 125, 254, 120, 229, 33, 175, 127, 194, 184:REM0B
530 DATA 127, 225, 205, 100, 2, 35, 126, 205, 100, 2:REMF0
540 DATA 35, 126, 205, 100, 2, 1, 0, 48, 205, 96:REMA0
550 DATA 0, 6, 64, 175, 197, 205, 214, 126, 193, 16:REMEB
560 DATA 248, 62, 85, 205, 214, 126, 6, 6, 33, 153:REM15
570 DATA 125, 126, 205, 214, 126, 35, 16, 249, 42, 149:REM46
580 DATA 125, 237, 91, 151, 125, 14, 0, 62, 60, 205:REMDF
590 DATA 214, 126, 62, 2, 205, 214, 126, 62, 164, 205:REM71
600 DATA 214, 126, 129, 79, 62, 64, 205, 214, 126, 129:REM21
610 DATA 79, 125, 205, 214, 126, 129, 79, 124, 205, 214:REME8
620 DATA 126, 129, 205, 214, 126, 14, 0, 62, 60, 205:REM88
630 DATA 214, 126, 62, 2, 205, 214, 126, 62, 249, 205:REM45
640 DATA 214, 126, 129, 79, 62, 64, 205, 214, 126, 129:REMF9
650 DATA 79, 123, 205, 214, 126, 129, 79, 122, 205, 214:REMC4
```



```
660 DATA 126, 129, 205, 214, 126, 229, 213, 235, 175, 237:REM54
670 DATA 82, 124, 254, 0, 40, 4, 6, 255, 24, 1:REM7F
680 DATA 69, 209, 225, 120, 254, 0, 202, 184, 126, 14:REM12
690 DATA 0, 62, 60, 205, 214, 126, 120, 205, 214, 126:REM19
700 DATA 125, 205, 214, 126, 129, 79, 124, 205, 214, 126:REM65
710 DATA 129, 79, 126, 205, 214, 126, 129, 79, 35, 16:REMDD
720 DATA 247, 121, 205, 214, 126, 195, 121, 126, 62, 120:REM5B
730 DATA 205, 214, 126, 58, 159, 125, 61, 202, 195, 127:REM71
740 DATA 50, 159, 125, 6, 4, 33, 0, 0, 43, 124:REM3C
750 DATA 181, 32, 251, 16, 249, 195, 160, 125, 229, 213:REM5E
760 DATA 197, 245, 6, 16, 16, 254, 14, 9, 87, 55:REMA0
770 DATA 205, 240, 126, 122, 7, 87, 13, 32, 247, 241:REME9
780 DATA 193, 209, 225, 201, 48, 27, 62, 5, 211, 255:REMD7
790 DATA 6, 8, 16, 254, 62, 6, 211, 255, 6, 8:REM27
800 DATA 16, 254, 62, 4, 211, 255, 6, 42, 16, 254:REM5D
810 DATA 253, 35, 201, 6, 63, 16, 254, 201, 60, 62:REM28
820 DATA 0, 60, 67, 72, 65, 82, 71, 69, 77, 69:REMC0
830 DATA 78, 84, 32, 82, 65, 80, 73, 68, 69, 32:REM82
840 DATA 66, 65, 83, 73, 67, 32, 32, 32, 40, 67:REM88
850 DATA 41, 32, 32, 32, 49, 53, 47, 49, 48, 47:REM7F
860 DATA 56, 50, 32, 32, 80, 65, 82, 32, 77, 73:REM79
870 DATA 67, 72, 69, 76, 32, 70, 79, 85, 82, 67:REM51
880 DATA 65, 68, 69, 32, 60, 113, 88, 127, 62, 4:REM2B
890 DATA 211, 255, 205, 205, 127, 254, 85, 32, 249, 6:REM35
900 DATA 6, 17, 64, 60, 205, 205, 127, 18, 19, 16:REMF5
910 DATA 249, 62, 32, 18, 19, 205, 205, 127, 254, 60:REM53
920 DATA 32, 38, 205, 205, 127, 71, 14, 0, 205, 205:REM8E
930 DATA 127, 111, 129, 79, 205, 205, 127, 103, 129, 79:REMA6
940 DATA 205, 205, 127, 119, 35, 129, 79, 16, 247, 205:REMCB
950 DATA 205, 127, 185, 32, 26, 62, 35, 18, 24, 210:REM65
960 DATA 254, 120, 32, 17, 62, 201, 50, 226, 65, 205:REM32
970 DATA 248, 1, 195, 25, 26, 69, 82, 82, 69, 85:REMC3
980 DATA 82, 33, 175, 127, 6, 6, 17, 156, 63, 126:REM9C
990 DATA 18, 35, 19, 16, 250, 205, 248, 1, 195, 114:REM37
1000 DATA 0, 60, 41, 205, 127, 197, 245, 62, 4, 211:REM69
1010 DATA 255, 14, 8, 219, 255, 23, 48, 251, 6, 37:REM7D
1020 DATA 16, 254, 62, 4, 211, 255, 6, 61, 16, 254:REM80
1030 DATA 253, 35, 219, 255, 71, 241, 203, 16, 23, 245:REMAD
1040 DATA 13, 32, 235, 241, 193, 201, 60, 3, 226, 65:REM10
1050 DATA 195, 88, 127, 120, 25, 26:REM73
```




```
AFF     EQU     28A7H               ;INITIALISATION
CLS     EQU     01C9H
CLAV    EQU     049H
ECROCT  EQU     264H
        ORG     7C44H
;************************************************
DEPART  LD      HL,(40A4H)          ;LECTURE
        LD      (DEBBAS),HL         ;DES
        LD      HL,(40F9H)          ;POINTEURS
        LD      (FINBAS),HL         ;BASIC
;-----------------------------------------------
NOM     CALL    CLS                 ;EFFACE ECRAN
        LD      HL,TITRE            ;AFFICHAGE
        CALL    AFF                 ;DU TITRE
        LD      DE,NOMPRO           ;LECTURE DU NOM
        LD      B,06H               ;DU PROGRAMME
        LD      HL,3D2CH            ;A ENREGISTRER
TOUCHE  PUSH    DE
ERRCLA  CALL    CLAV
        CP      20H
        JP      M,ERRCLA
        CP      5BH
        JP      P,ERRCLA
        POP     DE
        LD      (DE),A
        LD      (HL),A
        INC     DE
        INC     HL
        DJNZ    TOUCHE
;-----------------------------------------------
        LD      HL,MESG1            ;DEMANDE LE
        CALL    AFF                 ;NOMBRE DE
NBCOPI  LD      HL,3D95H            ;COPIES
        LD      (HL),20H
        CALL    CLAV
        CP      31H
        JP      M,NBCOPI
        CP      3AH
        JP      P,NBCOPI
        LD      HL,3D95H
        LD      (HL),A
        AND     0FH
        LD      (COPIES),A
;-----------------------------------------------
        LD      HL,MESG2            ;ATTENTE POUR
        CALL    AFF                 ;LA MISE EN
        CALL    CLAV                ;ROUTE DU
        JP      ENRG                ;MAGNETOPHONE
;**************** DIVERS MESSAGES *********************
TITRE   DEFM    'ENREGISTREMENT A 2000 BAUDS DES PROGRAMMES EN BASIC'
        DEFB    0DH
        DEFM    'VERSION 2.0  DU 15/10/82  PAR  MICHEL  FOURCADE'
        DEFB    0DH
        DEFW    0D0DH
        DEFM    'DONNEZ LE NOM DU PROGRAMME (6 CARACTERES): '
        DEFB    00H
MESG1   DEFW    0D0DH
        DEFM    'NOMBRE DE COPIES: '
        DEFB    00H
MESG2   DEFB    0DH
```

```
DEFM    'METTEZ LE MAGNETOPHONE EN MODE ENREGISTREMENT'
DEFB    0DH
        DEFM    'ET APPUYEZ SUR UNE TOUCHE .'
        DEFB    00H
;*******************************************************
DEBBAS  DEFW    0000H           ;RESERVATION DE
FINBAS  DEFW    0000H           ;MEMOIRES
NOMPRO  DEFM    '      '
COPIES  DEFB    00H
;*******************************************************
;PROGRAMME ENREGISTRANT LE PRECHARGEUR AU FORMAT "SYSTEM"
;QUI SERA ENSUITE CAPABLE DE LIRE LE PROGRAMME SUIVANT
;AU FORMAT RAPIDE "CRB"
;*******************************************************
ENRG    CALL    284H            ;ECRITURE DE L'AMORCE
        LD      A,55H           ;ECRITURE DE L'OCTET
        CALL    ECROCT          ;DE FORMAT "SYSTEM"
;------------------------------------------------------
        LD      B,06            ;ECRITURE DU NOM
        LD      HL,NOMPRO       ;DU PROGRAMME
ECRNOM  LD      A,(HL)
        INC     HL
        CALL    ECROCT
        DJNZ    ECRNOM
;------------------------------------------------------
        LD      HL,PRECHA       ;ECRIT LE PRECHARGEUR
CHARGE  LD      A,(HL)          ;AU FORMAT "SYSTEM"
        CP      3CH             ;LE BLOC DOIT COMMENCER
        JP      NZ,FINPRO       ;PAR L'OCTET 3C
        CALL    ECROCT          ;ECRIT 3C
;------------------------------------------------------
        INC     HL              ;ECRIT LA LONGUEUR
        LD      A,(HL)          ;DU BLOC
        LD      B,A
        CALL    ECROCT
;------------------------------------------------------
        XOR     A               ;ECRIT L'ADRESSE
        LD      C,A             ;D'IMPLANTATION
        LD      A,02            ;DU BLOC
        LD      E,A
LECADR  INC     HL
        LD      A,(HL)
        LD      D,A
        ADD     A,C             ;CALCULE LE
        LD      C,A             ;"CHECKSUM"
        LD      A,D
        CALL    ECROCT
        DEC     E
        JR      NZ,LECADR
;------------------------------------------------------
DONNEE  INC     HL              ;ECRIT TOUT LE
        LD      A,(HL)          ;BLOC DE DONNEES
        LD      D,A             ;AU FORMAT "SYSTEM"
        ADD     A,C             ;AVEC CALCUL
        LD      C,A             ;DU "CHECKSUM"
        LD      A,D
        CALL    ECROCT
        DJNZ    DONNEE
;------------------------------------------------------
        LD      A,C             ;ECRIT L'OCTET DE
        CALL    ECROCT          ;"CHECKSUM"
;------------------------------------------------------
        INC     HL              ;CHARGEMENT DU BLOC
        JP      CHARGE          ;SUIVANT
;------------------------------------------------------
FINPRO  CP      78H             ;APRES LE DERNIER BLOC
        PUSH    HL              ;IL DOIT Y AVOIR 78
        LD      HL,AFERR        ;SINON AFFICHER
        JP      NZ,ERRENR       ;ERREUR
        POP     HL
;------------------------------------------------------
        CALL    ECROCT          ;ECRIT L'OCTET 78
        INC     HL              ;ECRIT L'ADRESSE DE
        LD      A,(HL)          ;LANCEMENT DU PROGRAMME
        CALL    ECROCT          ;PRECHARGEUR QUI VIENT
        INC     HL              ;D'ETRE ENREGISTRE
        LD      A,(HL)          ;AU FORMAT "SYSTEM"
        CALL    ECROCT
;------------------------------------------------------
        LD      BC,3000H        ;DELAI ENTRE LE PRECHARG.
        CALL    0060H           ;ET LE PROGRAMME RAPIDE
;*******************************************************
;ECRITURE DES OCTETS CONSTITUANT LE PROGRAMME BASIC
;AU FORMAT RAPIDE "CRB"
;*******************************************************
        LD      B,40H           ;ECRITURE D'UNE AMORCE
AMORCE  XOR     A               ;DE 64 OCTETS 00
        PUSH    BC              ;AU FORMAT "CRB"
        CALL    SORTIE
        POP     BC
        DJNZ    AMORCE
;------------------------------------------------------
        LD      A,55H           ;ECRIT UN OCTET 55
        CALL    SORTIE          ;DE SYNCHRONISATION
        LD      B,06            ;ECRIT LE NOM
        LD      HL,NOMPRO       ;DU PROGRAMME
BCLNOM  LD      A,(HL)
        CALL    SORTIE
        INC     HL
```

```
        DJNZ    BCLNOM
;------------------------------------------------------
        LD      HL,(DEBBAS)     ;RECUPERE LES ADRESSES
        LD      DE,(FINBAS)     ;DE DEBUT ET FIN BASIC
        LD      C,00            ;"CHECKSUM" A ZERO
        LD      A,3CH           ;ECRIT L'OCTET 3C
        CALL    SORTIE          ;DE DEBUT DE BLOC
        LD      A,02            ;ECRIT UN BLOC
        CALL    SORTIE          ;DE LONGUEUR 2
        LD      A,0A4H          ;ECRIT L'ADRESSE
        CALL    SORTIE          ;D'IMPLANTATION DU BLOC
        ADD     A,C             ;C'EST-A-DIRE 40A4H
        LD      C,A
        LD      A,40H
        CALL    SORTIE
        ADD     A,C
        LD      C,A
        LD      A,L             ;ECRIT LE BLOC
        CALL    SORTIE          ;ADRESSE DEBUT DU BASIC
        ADD     A,C
        LD      C,A
        LD      A,H
        CALL    SORTIE
        ADD     A,C
        CALL    SORTIE          ;ECRIT LE "CHECKSUM"
;------------------------------------------------------
        LD      C,00            ;"CHECKSUM" A ZERO
        LD      A,3CH           ;ECRITURE D'UN AUTRE BLOC
        CALL    SORTIE
        LD      A,02            ;ECRIT LA LONGUEUR
        CALL    SORTIE          ;SOIT 2 OCTETS
        LD      A,0F9H          ;ECRIT L'ADRESSE
        CALL    SORTIE          ;D'IMPLANTATION
        ADD     A,C             ;DU BLOC: 40F9H
        LD      C,A             ;C'EST-A-DIRE
        LD      A,40H           ;POINTEUR DE FIN DE BASIC
        CALL    SORTIE
        ADD     A,C
        LD      C,A
        LD      A,E             ;ECRIT L'ADRESSE DE FIN
        CALL    SORTIE          ;DU PROGRAMME BASIC
        ADD     A,C             ;QUI VA ETRE CHARGE
        LD      C,A
        LD      A,D
        CALL    SORTIE
        ADD     A,C
        CALL    SORTIE          ;ECRITURE DU "CHECKSUM"
;------------------------------------------------------
COMPAR  PUSH    HL              ;CALCUL DE LA LONGUEUR
        PUSH    DE              ;DU BLOC A ENREGISTRER
        EX      DE,HL
        XOR     A               ;CETTE LONGUEUR SERA 0FFH
        SBC     HL,DE           ;JUSQU'AU DERNIER BLOC
        LD      A,H             ;A ENREGISTRER POUR
        CP      00              ;LEQUEL ELLE SERA
        JR      Z,LONG          ;LE NOMBRE EXACT
        LD      B,0FFH          ;D'OCTETS RESTANTS
        JR      SUITE
LONG    LD      B,L
SUITE   POP     DE
        POP     HL
        LD      A,B             ;SI LA LONGUEUR EST NULLE
        CP      00              ;C'EST QUE LE PROGRAMME
        JP      Z,TERM          ;EST ENREGISTRE
;------------------------------------------------------
        LD      C,00            ;"CHECKSUM" A ZERO
        LD      A,3CH           ;ECRITURE DE 3CH
        CALL    SORTIE          ;L'OCTET DE DEBUT DE BLOC
        LD      A,B             ;ECRIT LA LONGUEUR
        CALL    SORTIE          ;DU BLOC
        LD      A,L             ;ECRIT L'ADRESSE DU BLOC
        CALL    SORTIE
        ADD     A,C
        LD      C,A
        LD      A,H
        CALL    SORTIE
        ADD     A,C
        LD      C,A
BLOC    LD      A,(HL)          ;ECRITURE D'UN BLOC
        CALL    SORTIE          ;AU FORMAT "CRB"
        ADD     A,C             ;AVEC CALCUL
        LD      C,A             ;DU "CHECKSUM"
        INC     HL
        DJNZ    BLOC
        LD      A,C             ;ECRITURE DU "CHECKSUM"
        CALL    SORTIE
        JP      COMPAR          ;POUR LE BLOC SUIVANT
;------------------------------------------------------
TERM    LD      A,78H           ;ECRIT 78H APRES
        CALL    SORTIE          ;LE DERNIER BLOC
;------------------------------------------------------
        LD      A,(COPIES)      ;TESTE SI TOUTES LES
        DEC     A               ;COPIES DEMANDEES
        JP      Z,EXIT          ;SONT EFFECTUEES
        LD      (COPIES),A      ;COMPTEUR A JOUR
;------------------------------------------------------
        LD      B,04            ;DELAI AVANT LA
        LD      HL,0000         ;COPIE SUIVANTE
BCLO    DEC     HL
```

```
        LD      A,H
        OR      L
        JR      NZ,BCLO
        DJNZ    BCLO
        JP      ENRG                ;NOUVEL ENREGISTREMENT
;*******************************************************
;SOUS PROGRAMME D'ECRITURE D'UN OCTET AU FORMAT "CRB"
;*******************************************************
SORTIE  PUSH    HL                  ;SAUVEGARDES
        PUSH    DE
        PUSH    BC
        PUSH    AF
        LD      B,10H
        DJNZ    $
        LD      C,09H
        LD      D,A
        SCF
BCL1    CALL    ECRBIT
        LD      A,D
        RLCA
        LD      D,A
        DEC     C
        JR      NZ,BCL1
        POP     AF
        POP     BC
        POP     DE
        POP     HL
        RET
;*******************************************************
;SOUS PROGRAMME UTILISE PAR LE PRECEDENT POUR ECRIRE
;UN SEUL BIT
;*******************************************************
ECRBIT  JR      NC,TEMPO
        LD      A,05
        OUT     (255),A
        LD      B,08H
        DJNZ    $
        LD      A,06
        OUT     (255),A
        LD      B,08
        DJNZ    $
        LD      A,04
        OUT     (255),A
        LD      B,2AH
        DJNZ    $
        INC     IY
        RET
TEMPO   LD      B,3FH
        DJNZ    $
        RET
;*******************************************************
ENTOCT  EQU     7FCDH               ;ADRESSES DES DEUX
DEBUT   EQU     7F58H               ;SOUS PROGRAMMES
;*******************************************************
;PRECHARGEUR QUI SERA ENREGISTRE AU FORMAT "SYSTEM"
;EN DEBUT DE BANDE ET QUI PERMETTRA DE LIRE LE PROGRAMME
;AU FORMAT "CRB"
;*******************************************************
PRECHA  DEFW    3E3CH               ;3E=LONGUEUR 3C=DEB. BLOC
        DEFW    3C00H               ;ADRESSE D'IMPLANTATION
        DEFM    'CHARGEMENT RAPIDE BASIC   (C)   15/10/82  PAR MICHEL FOUR
CADE '
        DEFW    713CH               ;71=LONGUEUR 3C=DEB. BLOC
        DEFW    7F58H               ;ADRESSE D'IMPLANTATION
        LD      A,04
        OUT     (255),A
SYNCRO  CALL    ENTOCT              ;LIT UN OCTET "CRB"
        CP      55H                 ;ATTENTE DE L'OCTET
        JR      NZ,SYNCRO           ;55H DE SYNCHRO
        LD      B,06                ;LECTURE DES 6 OCTETS
        LD      DE,3C40H            ;DU NOM DU PROGRAMME
AFNOM   CALL    ENTOCT              ;AVEC AFFICHAGE
        LD      (DE),A              ;SUR L'ECRAN
        INC     DE
        DJNZ    AFNOM
        LD      A,20H               ;AFFICHE UN ESPACE
        LD      (DE),A
;-------------------------------------------------------
BCL2    INC     DE
        CALL    ENTOCT              ;LIT ET TESTE
        CP      3CH                 ;L'OCTET DE DEBUT
        JR      NZ,FINI             ;DE BLOC
        CALL    ENTOCT              ;LIT LA LONGUEUR
        LD      B,A                 ;DU BLOC
        LD      C,00                ;"CHECKSUM" A ZERO
        CALL    ENTOCT              ;LIT L'ADRESSE
        LD      L,A                 ;D'IMPLANTATION DU BLOC
        ADD     A,C                 ;AVEC CALCUL
        LD      C,A                 ;DU "CHECKSUM"
        CALL    ENTOCT
        LD      H,A
        ADD     A,C
        LD      C,A
BCL3    CALL    ENTOCT              ;LECTURE D'UN BLOC
        LD      (HL),A
        INC     HL
        ADD     A,C                 ;AVEC CALCUL
        LD      C,A                 ;DU "CHECKSUM"
        DJNZ    BCL3
        CALL    ENTOCT              ;LIT LE "CHECKSUM"
        CP      C                   ;ET COMPARE AVEC CELUI
        JR      NZ,ERREUR           ;CALCULE
        LD      A,23H               ;AFFICHE UN "#"
        LD      (DE),A              ;SUPPLEMENTAIRE POUR
        JR      BCL2                ;CHAQUE BLOC LU
;-------------------------------------------------------
FINI    CP      78H                 ;FIN DU PROGRAMME LU
        JR      NZ,ERREUR           ;SINON ERREUR
        LD      A,0C9H              ;PLACE UN "RET" POUR UN
        LD      (41E2H),A           ;DEMARRAGE AUTOIATIQUE
        CALL    1F8H                ;ARRET DU MOTEUR CASSETTE
        JP      1A19H               ;JUMP AU BASIC
;-------------------------------------------------------
AFERR   DEFM    'ERREUR'            ;MESSAGE ERREUR
ERREUR  LD      HL,DEBUT+57H        ;SOUS PROGRAMME AFFICHANT
ERRENR  LD      B,06                ;"ERREUR" EN CAS DE
        LD      DE,3F9CH            ;MAUVAISE LECTURE AU
XYZ     LD      A,(HL)              ;FORMAT "CRB"
        LD      (DE),A
        INC     HL
        INC     DE
        DJNZ    XYZ
EXIT    CALL    1F8H                ;ARRET MOTEUR
        JP      72H                 ;ET RETOUR AU BASIC
;*******************************************************
;SOUS PROGRAMME INCLUS DANS LE PRECHARGEUR PERMETTANT
;DE LIRE UN OCTET AU FORMAT "CRB"
;*******************************************************
        DEFW    293CH               ;29=LONGUEUR 3C=DEB. BLOC
        DEFW    7FCDH               ;ADRESSE D'IMPLANTATION
BCL4    PUSH    BC
        PUSH    AF
        LD      A,04
        OUT     (255),A
        LD      C,08H               ;COMPTEUR POUR 8 LECTURES
ATTBIT  IN      A,(255)             ;ATTENTE D'UN BIT
        RLA
        JR      NC,ATTBIT
        LD      B,25H               ;DELAI
        DJNZ    $
BCL5    LD      A,04                ;INITIALISATION
        OUT     (255),A             ;AVANT LECTURE
        LD      B,3DH               ;DELAI
        DJNZ    $
        INC     IY
        IN      A,(255)             ;LECTURE BIT
        LD      B,A                 ;SAUVEGARDE
        POP     AF
        RL      B                   ;ET ROTATION POUR
        RLA                         ;QUE "A" CONTIENNE
        PUSH    AF                  ;LES HUIT BITS APRES
        DEC     C                   ;LES HUIT LECTURES
        JR      NZ,BCL5             ;FIN DE BOUCLE
        POP     AF                  ;RECUPERATION DE L'OCTET
        POP     BC
        RET                         ;FIN DU SOUS PROGRAMME
;*******************************************************
;IMPLANTE DANS LES MEMOIRES 41E2H A 41E4H UN JUMP AU
;PRECHARGEUR
;*******************************************************
        DEFW    033CH               ;03=LONGUEUR 3C=DEB. BLOC
        DEFW    41E2H               ;ADRESSE D'IMPLANTATION
        JP      7F58H               ;AFIN QUE CE PROGRAMME
        DEFB    78H                 ;DEMARRE AUTOMATIQUEMENT
        DEFW    1A19H               ;POUR RETOURNER AU BASIC
        END     DEPART
```

Michel Fourcade

Améliorez la vitesse de vos enregistrements sur cassette

Cet article est composé de deux volets : le premier, de Roger Brousmiche, décrit une méthode pour compacter les enregistrements écrits et lus sur la cassette. Le second article dû à E. Ferrier, nous propose tout simplement de réduire l'amorce de chaque enregistrement. Libre à vous de cumuler les deux !

Voici quatre ans déjà, manuel en main, j'ai tapé sur le clavier de mon TRS80 flambant neuf :

```
10 DIM F(150)
20 garnissage du fichier
100 FOR I = TO 150:PRINT #-1,F(I)
110 NEXT
```

Rien de bien sorcier en somme, ...mais quelle déception lors de l'exécution ! Après quelques minutes, ne voyant rien venir à l'écran, j'ai tapé BREAK et ajouté le jalon :

```
105 PRINT I;
```

Et la lumière fut ! Un nouvel "I" apparaissait à l'écran toutes les quatre secondes, et il fallut plus de 10 minutes pour en lire "150", et par la même occasion remplir la cassette.

La raison est simple : chaque PRINT #-1 entraîne l'écriture d'une amorce de 256 octets 0 suivie de l'oc-

Programme D'ESSAI N° 1

```
1 '      FINUM1/BAS      (C) Roger Brousmiche et TRACE  FF
2 '      Configuration de base : modele 1 16K cassette  FE
3 '---------------------------------------------------  FD
10 ON ERROR GOTO 150    ' : CMD"T" en version disque  94
20 ON ERROR GOTO 0:REMF0
30 CLS : PRINT TAB(10) "FICHIER NUMERIQUE SUR CASSETTE (VERSION
1)" : PRINT:REM49
40 '     Programme d'essai  D8
50 CLEAR 1000 : K=150 : DIM F(K):REMBF
60 PRINT "GARNISSAGE DU FICHIER F(" ;K; ")":REMBB
70 FOR I=0 TO K : F(I)=I+1E5 : NEXT:REM72
80 PRINT "SAUVEGARDE SUR K7 DU FICHIER DE F( 0 ) A F(" ;K; ")":R
EM97
90 INPUT "K7 - RECORD - ENTER " ; ZZ$ : GOSUB 1000:REM21
100 PRINT "REBOBINEZ LA K7 -- REMISE A 0 DU FICHIER":REMD4
110 FOR I=0 TO K : F(I)=0 : NEXT:REMDB
120 PRINT "CHARGEMENT DU FICHIER K7 DE F( 0 ) A F(" ;K; ")":REM7
2
130 INPUT "K7 - PLAY - ENTER " ; ZZ$ : GOSUB 2000:REM81
140 PRINT "K7 -- STOP" : END:REMA2
150 RESUME NEXT:REM44
1000 A$=";"      ' Ecriture K7  5C
1010 FOR I=0 TO K : A$=A$+STR$(F(I))+";":REM1D
1020 IF LEN(A$)>235 AND I<K : A$=LEFT$(A$,LEN(A$)-1) : PRINT A$
: PRINT #-1,A$ : A$=";":REM46
1030 NEXT : A$=LEFT$(A$,LEN(A$)-1) : PRINT A$ : PRINT #-1,A$ : R
ETURN:REMBD
2000 J=-1   ' Lecture K7  0B
2010 INPUT #-1,A$ : PRINT A$:REM92
2020 FOR I=1 TO LEN(A$):REMDF
2030 IF MID$(A$,I,1)=";" : J=J+1 : F(J)=VAL(MID$(A$,I+1)):REMAC
2040 NEXT : IF J=K THEN RETURN ELSE 2010:REM59
```



Programme D'ESSAI N° 2

```
1 '      FINUM2/BAS       (C) Roger Brousmiche et TRACE  FF
2 '      Configuration de base : modele 1 16K cassette  FE
3 '------------------------------------------------------  FD
10 CLS : PRINT TAB(10) "FICHIER NUMERIQUE SUR CASSETTE (VERSION
2)":PRINT:REM5C
20 '     Initialisation  EC
30 CLEAR 50 : I=0 : J=0 : V=0 : B=0 : DIM EC%(16),LI%(16):REMD7
40 V=VARPTR(EC%(0)) : FOR I=0 TO 31 : READ J : POKE V+I,J : NEXT
:REMA7
50 V=VARPTR(LI%(0)) : FOR I=0 TO 31 : READ J : POKE V+I,J : NEXT
:REM90
60 '     Data ecriture K7  C4
70 DATA 205, 127, 10, 1, 0, 0, 217, 175, 205, 18, 2:REMEE
80 DATA 205, 135, 2, 217, 126, 205, 100, 2, 35, 121, 189:REME3
90 DATA 32, 247, 120, 188, 32, 243, 205, 248, 1, 201:REM65
100 '    Data lecture K7  9C
110 DATA 205, 127, 10, 1, 0, 0, 217, 175, 205, 18, 2:REMC6
120 DATA 205, 150, 2, 217, 205, 53, 2, 119, 35, 121, 189:REME5
130 DATA 32, 247, 120, 188, 32, 243, 205, 248, 1, 201:REM3D
140 '    Programme d'essai  74
150 K=151 : DIM F(K) : PRINT"GARNISSAGE D'UN FICHIER F(" ;K; ")"
:REM55
160 FOR I=0 TO K : F(I)=I+1E5 : NEXT:REM18
170 PRINT "SAUVEGARDE DU FICHIER SUR K7 DE F( 0 ) A F(" ;K-1; ")
":REM3E
180 INPUT "K7 -- RECORD -- ENTER " ; ZZ$ : V=VARPTR(EC%(0)) : GO
SUB 250:REMF3
190 PRINT "REBOBINEZ LA K7 -- REMISE A 0 DU FICHIER":REM7A
200 FOR I=0 TO K : F(I)=0 : NEXT:REM81
210 PRINT "CHARGEMENT DU FICHIER K7 DE F( 0 ) A F(" ;K-1; ")":RE
M19
220 INPUT "K7 -- PLAY -- ENTER " ; ZZ$ : V=VARPTR(LI%(0)) : GOSU
B 250:REM47
230 FOR I=0 TO K : PRINT F(I) ; : NEXT : PRINT : PRINT "K7 -- ST
OP" : END:REM83
240 '  Sous-routine ECRITURE/LECTURE K7  10
250 B=V : ON ERROR GOTO 280  : DEFUSR=V:REM23
260 ON ERROR GOTO 0 : V=VARPTR(F(K)) : IF V<0 : V=V+65536:REMF1
270 I=INT(V/256) : POKE B+4,V-256*I : POKE B+5,I : I=USR(VARPTR(
F(0))) : RETURN:REM02
280 IF V<0 : V=V+65536:REM6B
290 I=INT(V/256) : POKE 16526,V-I*256 : POKE 16527,I : RESUME NE
XT:REM1A
```



tet de synchronisation OA5H, ce qui prend 4 secondes à 500 bauds ou 62.5 octets/seconde.

Il faut donc s'arranger pour réduire au minimum le nombre d'instructions PRINT #-1, en remplissant au mieux les blocs de 248 caractères maximum de chaque enregistrement.

Pour ce faire on peut "concaténer" dans une variable A$, une suite de STR$ des éléments du fichier, en les séparant par un caractère non ambigu, par exemple un ";". La chaîne est écrite sur la cassette dès que le nombre de caractères dépasse 234. Et on recommence ce processus jusqu'à la fin du fichier.

Ce procédé, applicable aux fichiers à une dimension, est exploité dans le programme d'essai N° 1 ci-après. L'écriture cassette se fait en 52 sec. et la lecture en 66 sec. à 500 bauds (les temps doivent être divisés par 3 pour le modèle 3 à 1500 bauds).

On conçoit aisément que pour accélerer encore le processus, il suffit d'écrire tout le fichier sur cassette en un seul bloc, en y recopiant tout bonnement le contenu de la mémoire centrale. Ce n'est pas possible par l'instruction BASIC PRINT #-1.

On aura donc recours au langage machine en utilisant l'instruction VARPTR pour localiser le début et la fin du fichier, et à USR pour l'exécution de sous-routines binaires écrites dans des lignes DATA.

Ce procédé est exploité dans le programme d'essai N° 2 ci-après.

Les sous-routines binaires d'écriture/lecture casset-

te sont logées dans l'espace mémoire réservé pour deux fichiers "bidons" EC%(16) et LI%(16)... (pourquoi pas ?). Le DEFUSR est défini par : VARPTR(EC%(0)) à l'écriture ou : VARPTR(LI%(0)) à la lecture et l'exécution est obtenue par : I=USR(VARPTR(F(0))), après un POKE de l'adresse finale du fichier VARPTR(F(K)), K étant la DIM du fichier F( ) à copier.

Mais attention : du fait que la sous-routine se termine juste avant, le dernier indice du fichier ne sera pas sauvé. La DIM sera donc fixée à l'unité supérieure (très important).

L'écriture et la lecture s'effectuent en 14 sec. avec les mêmes paramètres que précédemment, soit un gain de temps (et de consommation de bandes) appréciable.

Cette méthode accepte aussi les fichiers à plusieurs dimensions. Par exemple dans un programme de "boursicotage", on souhaite sauver sur bande des suites de 100 cours successifs de 10 titres différents. Le fichier aura la configuration suivante : DIM F%(100,9), soit 10 colonnes (0 à 9) de 101 rangées (0 à 100).
On écrira dans la ligne 2010 : V=VARPTR(F%(100,9)) et dans la ligne 2020 : I=USR(VARPTR(F%(0,0))). Seul l'indice F%(100,10), dernière rangée de la dernière colonne ne sera pas sauvé sur la cassette.
Ce fichier comprend 2020 octets qui seront écrits ou lus en 36 secondes à 500 bauds. Ceci vaut bien 3 lignes de sous-routine... non ?

Ce programme tourne sur modèle 1, niveau 2 cassette ou disquette, et peut-être aussi sur le modèle 3, les CALL en direction de la ROM des sous-routines binaires étant compatibles (voir TRACE N° 2, différences entre 1 et 3).

Roger Brousmiche

Raccourcissez votre amorce

L'instruction PRINT #-1 qui commande l'écriture des enregistrements sur la cassette, fait précéder ceux-ci d'une amorce de 256 octets nuls et ce, pour une longueur d'enregistrement maximum de 255 octets. C'est dire qu'au mieux, le rendement est de 50 % (et beaucoup plus faible si l'on échange seulement quelques octets...).

L'objet du programme qui suit est de limiter cette amorce à quelques octets, le temps d'assurer le démarrage du moteur. Nous avons choisi 16 octets mais ce nombre peut être modifié au goût de chacun, après quelques essais, naturellement !

ATTENTION : ce programme est valable pour les "nouvelles ROMs" (celles qui affichent MEM SIZE). Pour les anciennes, le traitement du PRINT # s'effectue après celui du PRINT à et quelques corrections s'imposent.

Principe

L'instruction PRINT appelle une routine située en 206FH de la ROM. A cet emplacement, nous trouvons un CALL 41CAH qui est une sortie vers le DOS (quand il existe), laquelle sortie est "bouchée" par un RET en version cassette. C'est donc en 41CAH que nous placerons un branchement vers notre routine d'écriture d'une amorce de 16 octets.

Si l'on examine le contenu de la ROM après l'adresse 206FH, nous trouvons :

```
2072 : CP    '#'
2074 : JR    NZ,207CH
2076 : CALL  284H       ; écriture de l'amorce
2079 : ....             ; suite du traitement
```

Quant à la routine d'écriture de l'amorce, elle débute ainsi :

```
0284 : CALL  1FEH       ; démarre le moteur
0287 : LD    B,0FFH     ; longueur de l'amorce = 256
0289 : ....             ; suite du traitement
```

Notre routine aura donc la constitution suivante :

1) En assembleur :

```
         ORG   41CAH     ; interception du PRINT
         JP    MODIF
         ORG   4040H     ; emplacement libre en RAM
MODIF    POP   BC
         CP    '#"
         JP    NZ,207CH  ; ce n'est pas PRINT #
         CALL  01FEH
         LD    B,16      ; longueur amorce
         CALL  0289H
         JP    2079H
         END   6CCH      ; retour au BASIC
```

2) ou en BASIC :

```
1 CLEAR 50 : DIM B(11)
2 A$="*ROUTINE BINAIRE*"
3 V=VARPTR(A$) : A=PEEK(V+1)+256*PEEK(V+2)
4 POKE 16842,195 : POKE 16843,PEEK(V+1) : POKE 16844
,PEEK(V+2)
5 FOR I=0 TO 16 : READ B : POKE A+I,B : NEXT
6 DATA 193, 254, 35, 194, 124, 32, 205, 254, 1, 6
, 16, 205, 137, 2, 195, 121, 32
7 DELETE 3-7       ' FACULTATIF (REFAIRE RUN)
```

Il nous reste ensuite à faire un petit essai avec le programme ci-dessous :

```
10 CLS
20 INPUT "PREPARER CASSETTE POUR ENREGISTREMENT"; E$
30 FOR I=0 TO 11 : A=RND(10)-1+RND(0) : PRINT A, :
   PRINT # -1,A : NEXT
40 PRINT : INPUT "PREPARER CASSETTE POUR RELECTURE";
   E$
50 FOR I=0 TO 11 : INPUT#-1,B(I) : NEXT
60 FOR I=0 TO 11 : PRINT B(I), : NEXT
```

Dans le cas où vous utilisez des valeurs indicées, il y a tout de même intérêt à ne pas en "passer" une seule à la fois :

```
30 FOR I=0 TO 11 : A(I)=RND(10)-1+RND(0) : PRINT A(I),
   : NEXT
35 FOR I=0 TO 11 STEP 4 : PRINT #-1,A(I),A(I+1),A
   (I+2),A(I+3) : NEXT
50 FOR I=0 TO 11 STEP 4 : INPUT#-1,B(I),B(I+1),B(I+2)
   ,B(I+3) : NEXT
```

E. Ferrier

Sésame : ouvre-toi!



En général, qui possède un ordinateur individuel équipé de disquettes ne se soucie guère de protéger ses programmes par des mots de passe... qu'il risque d'oublier lui-même ! Mais, perfectionniste "sur les bords", il tend toujours à améliorer ou à adapter certains programmes du commerce et là, il risque de se heurter au cerbère qui a pour nom : PASSWORD !

Pour qui dispose du système d'exploitation NEWDOS-80 (version 2), le problème est résolu en quelques secondes : il suffit de taper "SYSTEM,1,AA=N"... et les mots de passe sont désactivés. Mais que faire si on n'a pas la chance de posséder NEWDOS-80 ? Très simple : fabriquer une "fausse clé", un mot de passe "bidon" qui sera accepté par l'ordinateur en lieu et place du vrai...

Les mots de passe sont codés sur 2 octets dans le Directory et il est possible de les lire au moyen d'un utilitaire de "dump" disquette (SUPERZAP par exemple).

Pour chaque programme, le Directory comporte deux lignes de 16 octets. Les mots de passe sont localisés au début de la seconde ligne : les deux premiers octets représentent le mot de passe de mise-à-jour du programme (update), les deux suivants le mot de passe d'accès (access).

Pour tout code introduit en hexadécimal (4 digits), le programme SESAME donne en 44 secondes une série de mots de passe fictifs de 3 ou 4 lettres. Le nombre moyen est de 7 mots différents puisqu'il y a 26 puissance 4 mots de 4 lettres et 16 puissance 4 codes possibles.

Il reste alors à choisir le moins rébarbatif et l'adjoindre au nom du programme protégé (sans oublier le séparateur ".") et l'ordinateur se laissera gentiment berner !

Exemple :

CODE ? 9FE9
CLES : RSPB, FOLH, LABM, RBRR, INLW

```
1 '------------------------------------------------------  FF
2 '      SESAME  - (C) 1983 Andre Beny et TRACE        !  FE
3 '      Configuration : modele 1 ou 3 disquette       !  FD
4 '------------------------------------------------------  FC
10 CLS : CLEAR 3000 : DEFINT A-Z : DEFFN X(X,Y)=X AND NOT Y OR Y
 AND NOT X :  C$="123456789ABCDEF" : U=26:REM17
20 DEFFN N(X)=16*INSTR(C$,MID$(H$,X,1))+INSTR(C$,MID$(H$,X+1,1))
:REM2F
30 DIM F1(U),F2(U),G1(U),G2(U) : FOR I=0 TO U : READ F1(I),F2(I)
,G1(I),G2(I) : NEXT:REMEF
40 INPUT "CODE ";H$ : X1=FN X(150,FN N(1)) : X2=FN X(66,FN N(3))
:REMD2
50 PRINT "CLES : "; : FOR D=0 TO U : FOR C=1 TO U:REM10
60 B=FN X(X1,FN X(F1(C),G1(D))) : IF B<97 OR B>122 THEN 90:REME4

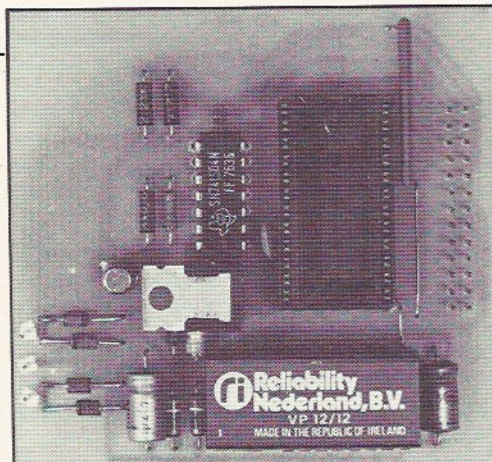
70 A=FN X(X2,FN X(F2(C),G2(D))) : IF A<97 OR A>122 THEN 90:REMDA

80 PRINT CHR$(A-32);CHR$(B-32);CHR$(C+64);CHR$(D+64+32*(D=0)); "
,";:REMCE
90 NEXT C,D : PRINT : GOTO 40:REM55
100 DATA 0, 0, 0, 0, 24, 69, 198, 25, 36, 32, 116, 174, 48, 3, 5
, 51, 92, 234, 49, 209, 72, 201, 64, 76, 116, 172, 242, 251:REM5
5
110 DATA 96, 143, 131, 102, 141, 110, 187, 47, 153, 77, 202, 178
, 165, 40, 120, 5, 177, 11, 9, 152:REM8F
120 DATA 221, 226, 61, 122, 201, 193, 76, 231, 245, 164, 254, 80
, 225, 135, 143, 205, 14, 119, 142, 194:REM8F
130 DATA 26, 84, 255, 95, 38, 49, 77, 232, 50, 18, 60, 117, 94,
251, 8, 151, 74, 216, 121, 10:REM5D
140 DATA 118, 189, 203, 189, 98, 158, 186, 32, 143, 127, 130, 10
5, 155, 92, 243, 244, 167, 57, 65, 67:REMAB
```

André Beny

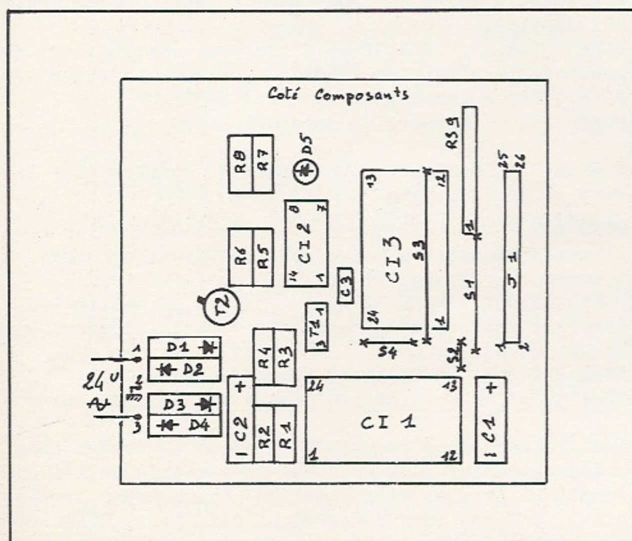


Ouvrez votre modèle 1 sur le monde extérieur (III)



Nous arrivons maintenant à l'étape de montage de la carte brûleur d'EPROM dont vous avez trouvé le circuit imprimé dans le numéro précédent (TRACE n° 6). Sont représentés ci-dessous le schéma électrique, une vue d'implantation des composants et une photo de la carte. Voici la liste des composants (toutes les résistances sont en 1/4 W) :

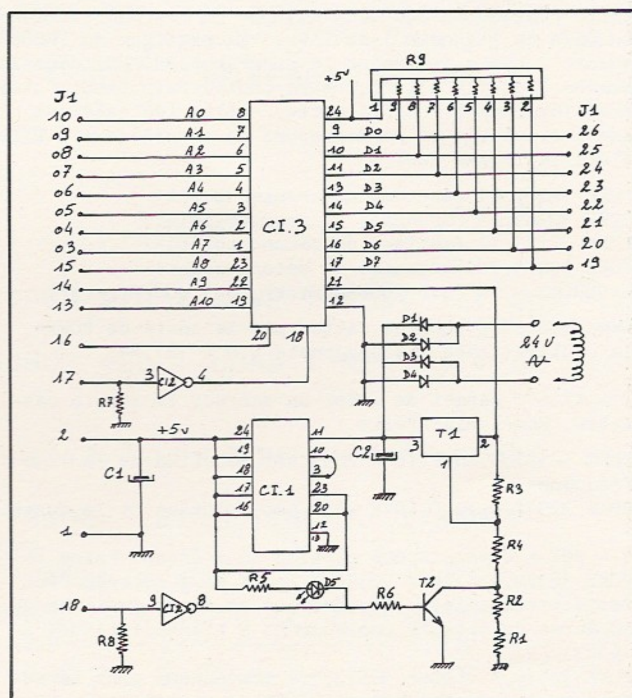
R1,R2 : 3,9K ohm
R3 : 470 ohm
R4 : 1,3K ohm
R5 : 220 ohm
R6,R7,R8 : 2,2 K ohm
R9 : réseau 8 x 10 K ohm SIL 9.8
C1 : 33 microfarad, 16 V
C2 : 4,7 microfarad, 63 V
C3 : 22 nanofarad
D1-D4 : diodes 1N4448
D5 : Led LD30A
T1 : régulateur LM317
T2 : transistor 2N2222
CI.1 : convertisseur 2 x 12 V - VP12.12
CI.2 : circuit intégré 74LS04
CI.3 : support 24 broches pour EPROM



Sur le support destiné à recevoir l'EPROM, il est possible de venir enficher un système spécialement conçu pour ce genre d'opération (non-représenté sur la photo) et qui permet un nombre important d'enfichages et de désenfichages des circuits sans risque d'usure des contacts (système à force d'insertion nulle) : il suffit simplement d'y poser l'EPROM (sans forcer) et d'actionner un petit levier qui vient refermer des contacts autour des 24 broches. Sur la photo et les schémas, on remarque un convertisseur 2 x 12 V qui permet de fabriquer du 24 volts (nécessaire à l'écriture de l'EPROM) à partir de la tension 5 V de la carte. C'est une solution idéale mais coûteuse ! Pour cette raison, nous avons prévu une entrée 24 volts en courant alternatif pour raccorder une alimentation extérieure (cette dernière solution est justifiée uniquement dans le cas où vous possédez déjà cette alimentation, car bien entendu, son prix est supérieur à celui du convertisseur).

La carte brûleur d'EPROM étant montée (ne pas oublier les 3 "straps" qui évitent l'emploi d'un circuit imprimé double face), il suffit de l'enficher sur la carte d'interface parallèle développée dans les numéros précédents, au moyen d'un petit connecteur latéral 26 contacts, ou en soudant des fils entre les deux cartes.

Dans le prochain numéro, nous aborderons enfin le logiciel qui permet de mettre en oeuvre cette extension. A bientôt !



Josué Bonifas

Cocktail d'extensions Basic

Une fois n'est pas coutume : ce programme s'adresse uniquement aux possesseurs de systèmes à cassette sur modèle 1. Vous n'êtes pas sans savoir qu'il existe des fonctions réservées au BASIC disque que l'on ne peut pas utiliser (enfin, en principe !) en niveau 2. Ce sont elles qui provoquent l'affichage du "L3 ERROR". Pourtant, leurs adresses d'interception sont bien réelles, et après quelques manipulations, j'en ai retrouvé 15 :

DEF	: 16732-16733	KILL	: 16786-16787
FIELD	: 16765-16766	SAVE	: 16801-16802
CLOSE	: 16774-16775	OPEN	: 16762-16763
NAME	: 16783-16784	PUT	: 16771-16772
RSET	: 16795-16796	MERGE	: 16780-16781
CMD	: 16756-16757	LSET	: 16792-16793
GET	: 16768-16769	LINE	: 16804-16805
LOAD	: 16777-16778		

Les autres fonctions utilisables sous DOS (FN, TIME$, CVI, CVS, etc...) provoque invariablement une SN ERROR.

Nous nous retrouvons donc avec 15 fonctions que nous allons pouvoir exploiter de la même manière qu'un USR, avec cependant l'impossiblité de passer des arguments. Qu'à cela ne tienne : nous les passerons par quelques POKEs !

Le programme que je vous propose vous permet d'appeler à tout moment 8 routines en langage machine grâce aux mots-clés des commandes inutilisées en niveau 2. Nous allons les examiner l'une après l'autre.

1 - USR (N) : permet le tracé instantané de lignes graphiques. En fait, je n'ai pas grand mérite pour cette routine puisqu'il s'agit de la retranscription en DATA de l'annexe 1 du livre "La pratique du TRS80", volume 2 (voir également le programme SEGMENT dans le numéro 5 de TRACE). Pour être tout-à-fait exact, j'ai très légèrement modifié cette routine (en espérant que les auteurs me pardonneront ce sacrilège !). Elle s'utilise comme ceci :

```
POKE 32267,X1 (abcisse du premier point)
POKE 32269,Y1 (ordonnée du premier point)
POKE 32271,X2 (abcisse du second point)
POKE 32273,Y2 (ordonnée du second point)
X=USR(n)      (n=0 pour effacer, n=1 pour tracer)
```

Remarque : si vous ne faites pas la série de POKEs, la routine trace une diagonale 0,0 - 127,47.

2 - KILL : permet de créer un son sur la sortie cassette. Pour cela, faire :

POKE 32545,F ou F (0 à 255) est fonction de la fréquence.
POKE 32547,D ou D (0 à 255) est fonction de la durée.

3 - PUT : blanchit une portion de l'écran. Faire :
POKE 32569,X : POKE 32570,Y (ou X et Y représentent respectivemnt les poids faibles et forts de l'adresse écran du premier emplacement à blanchir (15360 à 16383).
POKE 32572,X : POKE 32573,Y (même chose pour le second emplacement à blanchir. Cette possibilité permet le blanchissage d'un point sur 2 ou sur 3).
POKE 32575,R (ou R est le nombre de rangées (lignes horizontales) à blanchir).
POKE 32582,C (ou C est le nombre de colonnes-1 à blanchir).
POKE 32579,Z (ou Z est le code ASCII du caractère de blanchissage. En général, c'est 191).

4 - GET : permet d'inverser les caractères graphiques d'une zone quelconque de l'écran. Faire :

POKE 32603,X : POKE 32604,Y (ou X et Y sont les poids faibles et forts de l'adresse écran du premier emplacement à inverser).
POKE 32606,R (ou R est le nombre de rangées).
POKE 32609,C (ou C est le nombre de colonnes).

5 - OPEN : provoque le défilement vers le haut (scrolling) d'une partie quelconque de l'écran. Faire :

POKE 32641,X : POKE 32642,Y (adresse écran du premier emplacement à faire défiler. ATTENTION : le premier emplacement se trouve en fait sur la deuxième ligne du rectangle qui défile !).
POKE 32644,X : POKE 32645,Y (adresse écran du premier emplacement de la zone de destination. En général, cette adresse sera égale à la première diminuée de 64 (valeur d'une ligne). Une autre valeur provoquera un "scrolling" en diagonale !).
POKE 32647,R (ou R est le nombre de rangées moins 1).
POKE 32651,C (ou C est le nombre de colonnes).

6 - CLOSE : identique à OPEN mais le défilement a lieu vers le bas. Faire :

POKE 32673,X : POKE 32674,Y (adresse du dernier emplacement mémoire à faire défiler (se trouve sur l'avant-dernière ligne du rectangle qui défile).
POKE 32675,X : POKE 32676,Y (adresse de destination). Généralement, c'est l'adresse précédente augmentée de 64).
POKE 32679,R (ou R est le nombre de rangées moins 1).
POKE 32683,C (ou C est le nombre de colonnes).

7 - LOAD : identique à OPEN mais le défilement a lieu vers la gauche. Faire :

POKE 32705,X : POKE 32706,Y (adresse du premier emplacement à faire défiler (2ème colonne du rectangle qui défile).
POKE 32708,X : POKE 32709,Y (adresse destination. En général, il s'agit de l'adresse précédente moins 1).
POKE 32711,R (nombre de rangées).
POKE 32715,C (nombre de colonnes moins 1).

8 - SAVE : identique à OPEN mais le défilement a lieu vers la droite. Faire :

POKE 32737,X : POKE 32738,Y (adresse du dernier emplacement mémoire à faire défiler (avant dernière colonne du rectangle)).
POKE 32740,X : POKE 32741,Y (adresse de destination. Généralement, il s'agit de l'adresse ci-dessus augmentée de 1).
POKE 32743,R (ou R est le nombre de rangées).
POKE 32747,C (ou C est le nombre de colonnes moins 1).

Les six dernières fonctions décrites ci-dessus sont pré-programmées pour agir sur la totalité de l'écran. Ainsi, en faisant directemnt PUT (sans POKEs préalables), l'écran sera entièrement blanchi... OPEN, CLOSE, LOAD et SAVE déplaceront la totalité de l'écran

dans les 4 directions.
Au fait, il reste encore huit mots-clés inutilisés... Voilà ! J'espère que vous n'hésiterez pas à utiliser ces nouvelles possiblités qui, dans les jeux notamment, autorisent des effets que seul le langage machine pouvait apporter. Alors, la facilité de programmation du BASIC et la rapidité d'exécution du langage machine, ça mérite bien le détour !

```
1 REM -------------------------------------------------  39
2 REM ------ EXTENSIONS BASIC CASSETTE (16 K) --------  38
3 REM ------------- TRS-80 MODELE 1 -----------------  37
4 REM ----- (C) 1983 TRACE ET SCHLICK CHRISTOPHE -----  36
5 REM -------------------------------------------------  35
6 REM  34
10 POKE 16561,255 : POKE 16562,125 : CLEAR 50 : DEFINTA-Z : CLS:
REM76
20 PRINT@10,"COCKTAIL DE ROUTINES EN LANGAGE MACHINE":REM66
30 PRINT : PRINT:REM44
40 PRINT TAB(13) "USR(N) : TRACE DE LIGNES DROITES":REMF2
50 PRINT TAB(13) "KILL   : SON SUR SORTIE CASSETTE":REM0D
60 PRINT TAB(13) "PUT    : BLANCHISSAGE DE L'ECRAN":REM89
70 PRINT TAB(13) "GET    : INVERSION DE CARACTERES":REM3E
80 PRINT TAB(13) "OPEN   : DEFILEMENT VERS LE HAUT":REM25
90 PRINT TAB(13) "CLOSE  : DEFILEMENT VERS LE BAS ":REM33
100 PRINT TAB(13) "LOAD   : DEFILEMENT VERS LA GAUCHE":REMAC
110 PRINT TAB(13) "SAVE   : DEFILEMENT VERS LA DROITE":REM79
120 POKE 16526,000 : POKE 16527,126 : REM << USR(N) >>  62
130 POKE 16786,031 : POKE 16787,127 : REM << KILL   >>  43
140 POKE 16771,055 : POKE 16772,127 : REM << PUT    >>  3F
150 POKE 16768,089 : POKE 16769,127 : REM << GET    >>  22
160 POKE 16762,127 : POKE 16763,127 : REM << OPEN   >>  2B
170 POKE 16774,159 : POKE 16775,127 : REM << CLOSE  >>  16
180 POKE 16777,191 : POKE 16778,127 : REM << LOAD   >>  0A
190 POKE 16801,223 : POKE 16802,127 : REM << SAVE   >>  1C
200 PRINT : PRINT TAB(22) "ATTENTION !!":REM15
210 PRINT TAB(15) "CE MESSAGE S'AUTODETRUIRA":REM4D
220 PRINT TAB(17) "DANS, TRES EXACTEMENT":REM63
230 PRINT TAB(19) "QUINZE SECONDES...";:REME0
240 FOR A=32256 TO 32766 : READ B : POKE A,B : NEXT:REM50
250 FOR Z=0 TO 4999 : NEXT:REMC9
260 Z=USR(1) : FOR Z=0 TO 49 : NEXT : Z=USR(0):REMD9
270 FOR Z=0 TO 99 : ON RND(4) GOSUB 290,300,310,320 : NEXT:REM77

280 PUT : FOR Z=0 TO 49 : NEXT : GET : GOTO 330:REM02
290 OPEN : RETURN:REM6F
300 CLOSE : RETURN:REM61
310 LOAD : RETURN:REM56
320 SAVE : RETURN:REM46
330 CLS : PRINT " IMPRESSIONNANT " : KILL:REME3
340 B=1 : FOR Y=0 TO 7:REM56
350 A=1 : B=B+4 : FOR X=0 TO 45 : GOSUB 420 : SAVE : NEXT X : CL
OSE:REM47
360 A=1 : B=B+4 : FOR X=0 TO 45 : GOSUB 420 : LOAD : NEXT X : CL
OSE:REM43
370 NEXTY:REMAD
380 PUT : PRINT @216," MAINTENANT ";@343," LES ROUTINES ";:REM9D

390 PRINT @469," SONT UTILISABLES ";@600," DIRECTEMENT ";:REMFF
400 PRINT @728," EN BASIC... ";:REMB2
410 IF INKEY$="" THEN 410 ELSE CLS : STOP:REM74
420 POKE 32545,A : POKE 32547,B : A=A+1 : KILL : RETURN:REM9D
430 REM -------------------------------------------------  8B
440 REM ----------- ROUTINES SOUS FORME DE DATA --------  81
450 REM -------------------------------------------------  77
460 REM ---- FONCTION PLOT : (cf A. PINAUD & P. GIRAUD)-  6D
470 DATA 253,33,255,127,205,127,10:REMD1
```



```
480 DATA 253,117,0,6,0,14,0,22,127,30,47:REMB5
490 DATA 237,67,28,127,205,13,127,221,33,24,127,120,38,1:REM85
500 DATA 186,56,2,38,255,121,187,46,1,56,2,46,255,34,24,127:REMD
3
510 DATA 120,146,242,55,126,237,68,103,121,147,242,63,126:REM38
520 DATA237,68,111,34,26,127,124,189,56,52,203,47,50,30,127:REM9
3
530 DATA 37,248,58,25,127,221,134,5,50,29,127,58,30,127,133:REMB
9
540 DATA50,30,127,221,190,3,56,15,221,150,3,50,30,127,58,28:REM9
A
550 DATA127,221,134,0,50,28,127,237,75,28,127,205,13,127,24:REM8
1
560 DATA 209,125,203,47,50,30,127,45,248,58,24,127,221,134:REMD4

570 DATA4,50,28,127,58,30,127,132,50,30,127,221,190,2,56,15:REM7
7
580 DATA 221,150,2,50,30,127,58,29,127,221,134,1,50,29,127:REMD1

590 DATA 237,75,28,127,205,13,127,24,209,62,1,24,2,62,2,229:REM8
A
600 DATA 213,197,245,197,121,183,31,71,120,128,128,185,40,5:REM7
3
610 DATA 56,3,5,24,245,80,193,120,183,31,95,106,38,0,41,41:REMAA

620 DATA 41,41,41,41,121,146,146,146,135,128,147,147,71,22:REM9D

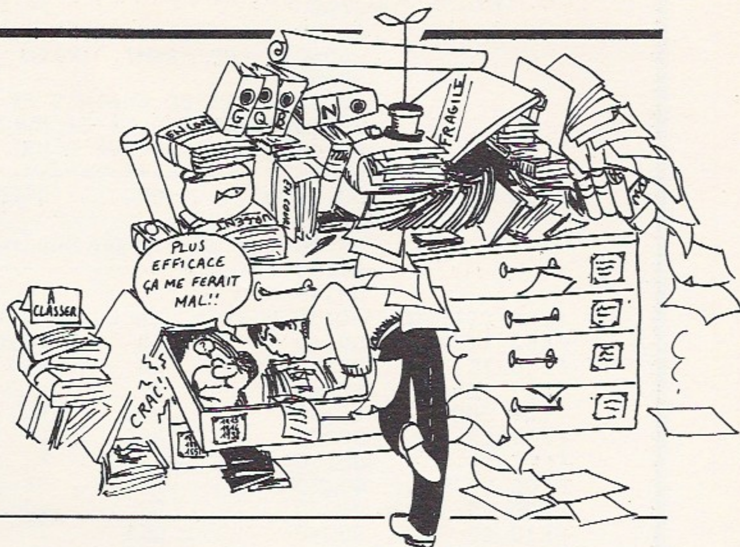
630 DATA 0,25,17,0,60,25,126,183,250,241,126,62,128,119,175:REM6
5
640 DATA 184,62,1,40,4,135,5,32,252,71,241,254,1,32,4,126:REMCB
650 DATA 176,24,3,120,47,166,119,193,209,225,201,253,126,0:REM7C

660 DATA 183,202,180,126,205,176,126,201,0,0,0,0,0,0,0:REM67
670 REM ------- FONCTION SOUND -------------------------  9A
680 DATA 217,38,0,46,127,68,62,1,211,255,16,254:REM78
690 DATA 68,62,2,211,255,16,254,45,32,239,217,201:REM0D
700 REM ------- FONCTION WHTSCR ------------------------  7C
710 DATA 217,33,0,60,17,1,60,62,16,213,229:REM61
720 DATA 54,191,1,63,0,237,176,225,209,1,64,0:REMBB
730 DATA 235,237,74,235,237,74,61,32,233,217,201:REM11
740 REM ------- FONCTION INVERT ------------------------  54
750 DATA 217,33,0,60,22,16,229,6,64,126,254,32,32,2:REM7E
760 DATA 198,96,203,127,40,4,47,198,64,119,35,16:REME1
770 DATA 238,225,1,64,0,237,74,21,32,226,217,201:REMF8
780 REM ------- FONCTION SCROLL UP ---------------------  2B
790 DATA 217,33,64,60,17,0,60,62,15,213,229:REMD8
800 DATA 1,64,0,237,176,225,209,1,64,0,235:REMFF
810 DATA 237,74,235,237,74,61,32,235,217,201:REM84
820 REM ------- FONCTION SCROLL DOWN -------------------  03
830 DATA 217,33,191,63,17,255,63,62,15,213,229:REM0D
840 DATA 1,64,0,237,184,225,209,1,64,0,235:REMD8
850 DATA 237,66,235,237,66,61,32,235,217,201:REM5A
860 REM ------- FONCTION SCROLL LEFT -------------------  DB
870 DATA 217,33,1,60,17,0,60,62,16,213,229:REMC0
880 DATA 1,63,0,237,176,225,209,1,64,0,235:REMB0
890 DATA 237,74,235,237,74,61,32,235,217,201:REM34
900 REM ------- FONCTION SCROLL RIGHT ------------------  B3
910 DATA 217,33,254,63,17,255,63,62,16,213,229:REMBC
920 DATA 1,63,0,237,184,225,209,1,64,0,235:REM89
930 DATA 237,66,235,237,66,61,32,235,217,201:REM0A
940 REM --------------- OUF, C'EST FINI !! -------------  8B
```



Christophe Schlick

Un tri encore plus efficace



Vous vous souvenez sûrement du programme de " tri efficace" exposé par Roger Buvat dans notre numéro 3. Aujourd'hui, l'auteur nous propose une version améliorée permettant le tri multi-clés et multi-tableaux. Le luxe, quoi !

En attendant que les microprocesseurs de 16 bits et les mémoires de 128 K et plus se généralisent dans nos ordinateurs individuels, le présent programme nous permettra le tri direct de tous les tableaux constituant le fichier (si celui-ci tient dans la mémoire centrale), en fonction de critères multiples et en ordre ascendant ou descendant pour chacun d'eux. Dans son principe, cette routine dérive de la précédente mais porte sur un tableau virtuel résultant du chaînage des tableaux réels composant les enregistrements. Il n'y a donc pas de limitation propre au nombre des tableaux susceptibles d'être triés, ni au nombre des clés puisque chaque tableau est une clé potentielle dans l'ordre des paramètres passés.

L'appel de cette routine en BASIC consiste en une succession d'instructions USR (paramètre). Pour simplifier un peu l'écriture, on pourra utiliser la forme U=USR(param1) OR USR(param2) or USR(param3)... Le premier paramètre doit indiquer le numéro de l'enregistrement du début du tri. Si celui-ci doit commencer au numéro 1 (cas général), il est inutile de fournir USR(1) qui est implicite par défaut. Les paramètres des USR suivants seront VARPTR(Variable(Ø)) concernant dans l'ordre la clé primaire, la clé secondaire, la clé tertiaire, etc... jusqu'au dernier tableau impliqué par le tri des enregistrements. Normalement ascendant, le sens du tri peut être inversé pour une ou plusieurs clés, en faisant suivre chacune d'elle par USR(1). Enfin le dernier paramètre à passer, celui qui déclenchera le tri, doit être le numéro du dernier enregistrement que l'on veut trier. Si celui-ci se confond avec la fin des enregistrements définis (cas général), terminer tout simplement la série par USR(Ø). Dans l'exemple cité plus haut, pour le tri normal complet du répertoire d'adresses, la ligne BASIC de commande sera constituée par :

```
U=USR(VARPTR(VILLE$(Ø))) OR USR(VARPTR(NOM$(Ø))) OR
USR(VARPTR(PRENOM$(Ø))) OR USR(VARPTR(ADRESSE$(Ø)))
OR USR(Ø)
```

En insérant USR(1) entre les deux premiers USR, le tri sera en ordre descendant pour la VILLE et ascendant pour les clés suivantes. Si le paramètre de fin de tri s'avérait supérieur au numéro du dernier enregistrement défini, c'est cette dernière valeur qui serait prise en compte. Au cas où le paramètre de début de tri serait égal ou supérieur au paramètre de fin de tri, celui-ci n'aurait évidemment pas lieu. Naturellement, les instructions USR doivent être précédés par DEFUSR sous DISK BASIC ou par les POKE 16 526 et 16527 en BASIC Level 2, pour le branchement à l'adresse du début de la routine.

Les contraintes d'emploi sont assez classiques :
- Tableaux de chaînes de caractères à une seule dimension.
- Même dimension pour tous les tableaux impliqués dans le tri.
- Les éléments d'indice Ø sont réservés, car ils sont utilisés par la routine pour le chaînage des tableaux.
- Si l'on utilise des nombres (sous forme de chaîne de caractère ASCII) comme clés, il faut veiller à observer la même position du point décimal à partir de la gauche pour tous les éléments, en complétant par des espaces ou des zéros si nécessaire.

Remarquons que les éléments d'un même tableau peuvent avoir des longueurs différentes et même nulles. Une recherche étant faite immédiatement avant le début du tri pour déterminer la longueur utile de tableau (limitée par la rencontre éventuelle d'un élément non défini), le dimensionnement des tableaux lors de leur déclaration par DIM peut donc être supérieur à l'effectif des éléments réellement définis. Pour la comparaison des clés, la hiérarchie attribuée est la suivante :

- élément vide (longueur nulle) ;
- codes ASCII 2ØH (espace) à 4ØH (escargot) ;
- codes des lettres 41H à 5FH ou 61H à 7FH ;
- code 61H (variable suivant les générateurs de caractères) ;
- codes graphiques 8ØH à FFH.

Ainsi l'ordre alphabétique est toujours respecté, que les lettres soient en majuscules ou en minuscules, ce qui évite les classements tels que DUPONT - DURAND - Dubois obtenus avec GSF. Entre lettres, chiffres et autres caractères, la hiérarchie est certes discutable mais sans doute préférable à un ordre aléatoire comme dans un tri avec ENHBAS.

Comparativement à la routine du numéro 3, celle-ci est un peu moins rapide pour le tri d'index. C'est évidemment la rançon d'une plus grande sophistication. Mais elle reste cependant encore supérieure dans ce domaine aux routines de GSF et de ENHBAS pourtant réputées.

```
                00010 ; SUPER TRI SHELL-METZNER
                00020 ;
                00030 ; AUTEUR : ROGER BUVAT
                00040 ; COPYRIGHT "TRACE" ET L'AUTEUR
                00050 ;
                00060 ; TRI DE CHAINES DE CARACTERES
                00070 ; MULTI-CLES ET MULTI-TABLEAUX
                00080 ; EN TABLEAUX A UNE SEULE DIMENSION
                00090 ; ==========================================================
FE00            00100         ORG     0FE00H          ; * POUR 48 K DE RAM
                00110
                00120 ; ACQUISITION DES PARAMETRES
                00130 ; --------------------------
                00140
FE00 CD7F0A     00150 START   CALL    0A7FH           ; ENTREE DANS HL
FE03 110000     00160         LD      DE,0000         ; STO VARPTR
FE06 7C         00170         LD      A,H
FE07 FE40       00180         CP      40H
FE09 3009       00190         JR      NC,VARPTR       ; SI ENTREE >= 16384
FE0B 15         00200         DEC     D
FE0C 14         00210         INC     D
FE0D 2039       00220         JR      NZ,SUITE        ; SI PAS 1ERE ENTREE
FE0F B5         00230         OR      L
FE10 C8         00240         RET     Z               ; SI 1ERE ENTREE = 0
FE11 2B         00250         DEC     HL
FE12 182C       00260         JR      DEBUT
FE14 2204FE     00270 VARPTR  LD      (START+4),HL
FE17 7A         00280         LD      A,D
FE18 B3         00290         OR      E
FE19 200C       00300         JR      NZ,CHAINE       ; SI PAS 1ER VARPTR
FE1B 22ACFE     00310         LD      (TRI3+6),HL
FE1E 010000     00320         LD      BC,0000         ; STO #DEBUT - 1
FE21 09         00330         ADD     HL,BC
FE22 09         00340         ADD     HL,BC
FE23 09         00350         ADD     HL,BC
FE24 117DFE     00360         LD      DE,TRI1-3
FE27 EB         00370 CHAINE  EX      DE,HL
FE28 23         00380         INC     HL
FE29 73         00390         LD      (HL),E
FE2A 23         00400         INC     HL
FE2B 72         00410         LD      (HL),D
FE2C C9         00420         RET
FE2D 2AACFE     00430 RAZ     LD      HL,(TRI3+6)
FE30 AF         00440         XOR     A
FE31 77         00450         LD      (HL),A
FE32 23         00460         INC     HL
FE33 5E         00470         LD      E,(HL)
FE34 77         00480         LD      (HL),A
FE35 23         00490         INC     HL
FE36 56         00500         LD      D,(HL)
FE37 77         00510         LD      (HL),A
FE38 7A         00520         LD      A,D
FE39 B3         00530         OR      E
FE3A EB         00540         EX      DE,HL
FE3B 20F3       00550         JR      NZ,RAZ+3        ; SI PAS FINI
FE3D 2204FE     00560         LD      (START+4),HL
FE40 221FFE     00570 DEBUT   LD      (VARPTR+11),HL
FE43 C9         00580         RET
FE44 3E3F       00590 INVERS  LD      A,3FH           ; CODE CCF
FE46 12         00600         LD      (DE),A
FE47 C9         00610         RET
FE48 B7         00620 SUITE   OR      A
FE49 2008       00630         JR      NZ,FIN          ; SI H > 0
FE4B 2D         00640         DEC     L
FE4C 28F6       00650         JR      Z,INVERS        ; SI ENTREE = 1
FE4E 2C         00660         INC     L
FE4F 2002       00670         JR      NZ,FIN          ; SI ENTREE <> 0
FE51 2640       00680         LD      H,40H           ; HL=16384
FE53 E5         00690 FIN     PUSH    HL
FE54 EB         00700         EX      DE,HL           ; HL=DERNIER VARPTR
FE55 2B         00710         DEC     HL
FE56 46         00720         LD      B,(HL)
FE57 2B         00730         DEC     HL
FE58 4E         00740         LD      C,(HL)
FE59 0B         00750         DEC     BC
FE5A 110400     00760         LD      DE,0004
FE5D 19         00770         ADD     HL,DE
FE5E 5A         00780         LD      E,D             ; DE=0000
FE5F 23         00790 EXPLOR  INC     HL
FE60 23         00800         INC     HL
FE61 23         00810         INC     HL
```

```
FE62 7E                LD      A,(HL)
FE63 B7                OR      A
FE64 2806              JR      Z,FEXPL         ; SI ELEMENT NON DEFINI
FE66 13                INC     DE
FE67 0B                DEC     BC
FE68 78                LD      A,B
FE69 B1                OR      C
FE6A 20F3              JR      NZ,EXPLOR       ; SI PAS FINI
FE6C E1        FEXPL   POP     HL
FE6D DF                RST     24
FE6E 3801              JR      C,$+3           ; SI #FIN < DIM TABLEAU
FE70 EB                EX      DE,HL
FE71 ED5B1FFE          LD      DE,(VARPTR+11)
FE75 AF                XOR     A               ; CARRY = 0
FE76 ED52              SBC     HL,DE
FE78 38B3              JR      C,RAZ           ; SI #FIN < #DEBUT
FE7A EB                EX      DE,HL

               ; ROUTINE DE TRI
               ; --------------

FE7B 42                LD      B,D             ; (N DANS DE)
FE7C 4B                LD      C,E             ; H=N (H DANS BC)
FE7D 210000            LD      HL,0000         ; STO PTR DE DEBUT DU TRI
FE80 78        TRI1    LD      A,B
FE81 CB3F              SRL     A
FE83 47                LD      B,A
FE84 79                LD      A,C
FE85 1F                RRA
FE86 4F                LD      C,A             ; H=INT(H/2) (H DANS BC)
FE87 B0                OR      B
FE88 28A3              JR      Z,RAZ           ; SI TRI TERMINE
FE8A E5                PUSH    HL
FE8B DDE1              POP     IX
FE8D E5                PUSH    HL              ; SAUVE PTR DE TABLEAU
FE8E 09                ADD     HL,BC
FE8F 09                ADD     HL,BC
FE90 09                ADD     HL,BC
FE91 E5                PUSH    HL
FE92 FDE1              POP     IY
FE94 210000            LD      HL,0000         ; I=0
FE97 D5        TRI2    PUSH    DE              ; SAUVE N
FE98 23                INC     HL              ; I=I+1
FE99 E5                PUSH    HL              ; SAUVE I
FE9A 110300            LD      DE,0003
FE9D DD19              ADD     IX,DE
FE9F FD19              ADD     IY,DE
FEA1 DDE5              PUSH    IX              ; SAUVE PTR DE E$(I)
FEA3 FDE5              PUSH    IY              ; SAUVE PTR DE E$(I+H)
FEA5 2B                DEC     HL              ; J-1=I-1
FEA6 D9        TRI3    EXX
FEA7 DDE5              PUSH    IX
FEA9 FDE5              PUSH    IY
FEAB 210000            LD      HL,0000         ; STO VARPTR CLE PRIMAIRE
FEAE 2237FF            LD      (STEP+1),HL
FEB1 7E        TRI4    LD      A,(HL)
FEB2 32EDFE            LD      (TEST),A
FEB5 CD59FF            CALL    LOAD
FEB8 79                LD      A,C
FEB9 B7                OR      A
FEBA 78                LD      A,B
FEBB 2003              JR      NZ,TRI6         ; SI (IY+00) <> 0
FEBD B9        TRI5    CP      C
FEBE 1823              JR      TRI8+1
FEC0 B7        TRI6    OR      A
FEC1 28FA              JR      Z,TRI5          ; SI (IX+00) = 0 = (IY+00)
FEC3 7E        TRI7    LD      A,(HL)
FEC4 FE61              CP      61H
FEC6 1A                LD      A,(DE)
FEC7 3008              JR      NC,MINUSC
FEC9 FE61              CP      61H
FECB 380A              JR      C,COMPAR        ; SI TOUT EN MAJUSCULES
FECD D620              SUB     20H
FECF 1806              JR      COMPAR
FED1 FE61      MINUSC  CP      61H
FED3 3002              JR      NC,COMPAR       ; SI TOUT EN MINUSCULES
FED5 C620              ADD     A,20H
FED7 BE        COMPAR  CP      (HL)
FED8 2013              JR      NZ,TEST         ; SI E$(J) <> E$(J+H)
FEDA 0D                DEC     C
FEDB 2805              JR      Z,TRI8          ; SI B >= C
FEDD 13                INC     DE
```

```
FEDE 23         01640          INC    HL
FEDF 10E2       01650          DJNZ   TRI7
FEE1 37         01660          SCF
FEE2 05         01670 TRI8     DEC    B
FEE3 2008       01680          JR     NZ,TEST           ; SI B > C
FEE5 CD36FF     01690          CALL   STEP
FEE8 2A37FF     01700          LD     HL,(STEP+1)
FEEB 20C4       01710          JR     NZ,TRI4           ; SI PAS DERNIER TABLEAU
FEED 00         01720 TEST     NOP                      ; POUR CCF SI TRI INVERSE
FEEE 3014       01730          JR     NC,PERMUT
FEF0 D9         01740          EXX
FEF1 E1         01750 TRI9     POP    HL
FEF2 E1         01760          POP    HL
FEF3 FDE1       01770          POP    IY                ; RESTAURE PTR DE E$(I+H)
FEF5 DDE1       01780          POP    IX                ; RESTAURE PTR DE E$(I)
FEF7 E1         01790          POP    HL                ; RESTAURE I
FEF8 D1         01800          POP    DE                ; RESTAURE N
FEF9 E5         01810          PUSH   HL
FEFA 09         01820          ADD    HL,BC             ; I+H
FEFB ED52       01830          SBC    HL,DE             ; I+H-N
FEFD E1         01840          POP    HL
FEFE 2097       01850          JR     NZ,TRI2           ; SI I+H-N <> 0
FF00 E1         01860          POP    HL                ; RESTAURE PTR DE TABLEAU
FF01 C380FE     01870          JP     TRI1
FF04 CD59FF     01880 PERMUT   CALL   LOAD
FF07 DD7402     01890          LD     (IX+02),H
FF0A DD7501     01900          LD     (IX+01),L
FF0D FD7202     01910          LD     (IY+02),D
FF10 FD7301     01920          LD     (IY+01),E
FF13 FD7000     01930          LD     (IY+00),B
FF16 DD7100     01940          LD     (IX+00),C
FF19 CD36FF     01950          CALL   STEP
FF1C 20E6       01960          JR     NZ,PERMUT         ; SI PAS DERNIER TABLEAU
FF1E D9         01970          EXX
FF1F ED42       01980          SBC    HL,BC             ; J-1-H
FF21 38CE       01990          JR     C,TRI9            ; POURSUITE DU CYCLE SHELL
FF23 EB         02000          EX     DE,HL
FF24 E1         02010          POP    HL
FF25 E1         02020          POP    HL
FF26 E5         02030          PUSH   HL
FF27 FDE1       02040          POP    IY                ; NOUVEAU PTR DE E$(J+H)
FF29 ED42       02050          SBC    HL,BC
FF2B ED42       02060          SBC    HL,BC
FF2D ED42       02070          SBC    HL,BC
FF2F E5         02080          PUSH   HL
FF30 DDE1       02090          POP    IX                ; NOUVEAU PTR DE E$(J)
FF32 EB         02100          EX     DE,HL
FF33 C3A6FE     02110          JP     TRI3              ; RETOUR ARRIERE (METZNER)
FF36 210000     02120 STEP     LD     HL,0000           ; STO VARPTR CLE COURANTE
FF39 23         02130          INC    HL
FF3A 5E         02140          LD     E,(HL)
FF3B 23         02150          INC    HL
FF3C 56         02160          LD     D,(HL)
FF3D 7A         02170          LD     A,D
FF3E B3         02180          OR     E
FF3F C8         02190          RET    Z                 ; SI DERNIER TABLEAU
FF40 2AACFE     02200          LD     HL,(TRI3+6)
FF43 EB         02210          EX     DE,HL
FF44 2237FF     02220          LD     (STEP+1),HL
FF47 ED52       02230          SBC    HL,DE
FF49 EB         02240          EX     DE,HL
FF4A C1         02250          POP    BC
FF4B FDE1       02260          POP    IY
FF4D DDE1       02270          POP    IX
FF4F DDE5       02280          PUSH   IX
FF51 FDE5       02290          PUSH   IY
FF53 C5         02300          PUSH   BC
FF54 DD19       02310          ADD    IX,DE
FF56 FD19       02320          ADD    IY,DE
FF58 C9         02330          RET
FF59 FD6602     02340 LOAD     LD     H,(IY+02)
FF5C FD6E01     02350          LD     L,(IY+01)
FF5F DD5602     02360          LD     D,(IX+02)
FF62 DD5E01     02370          LD     E,(IX+01)
FF65 DD4600     02380          LD     B,(IX+00)
FF68 FD4E00     02390          LD     C,(IY+00)
FF6B C9         02400          RET
0000            02410          END
00000 TOTAL ERRORS
30295  TEXT AREA BYTES LEFT
```

Roger Buvat

Lignes vitales

La définition d'une ligne vitale pourrait être celle-ci : "ligne dont la concaténation (la fusion) avec la ligne de numéro immédiatement inférieur entraînerait une ou plusieurs anomalies de fonctionnement".

Voici un petit exemple. Le programme ci-dessous effectue la mise à zéro des éléments du tableau A(i) sans utilisation d'une boucle FOR/NEXT :

```
10 DEFINT A-Z : N=25 : DIM A(N)
20 I=1
30 A(I)=0
40 I=I+1
50 IF I<=N THEN GOTO 30
60 END
```

Nous remarquons que la concaténation de la ligne 30 avec celle de numéro immédiatement inférieur (ici 20), entraînerait un "UL? ERROR" en ligne 50. Les lignes 20, 40 et 50, quant à elles, peuvent être fusionnées, ce qui donnerait :

```
10 DEFINT A-Z : N=25 : DIM A(N) : I=1
30 A(I)=0 : I=I+1 : IF I<=N THEN GOTO 30
60 END
```

Par l'exemple précédent, nous pouvons "sentir" intuitivement l'existence pour tout programme BASIC d'un "noyau minimal" composé de l'ensemble des lignes vitales de ce programme. En nous référant au programme ci-dessus, nous pouvons raisonnablement penser avoir trouvé le noyau minimal composé des lignes 10, 30 et 60. Pourtant, il n'en est rien car la ligne 60 peut être fusionnée à la ligne 30 qui devient alors :

```
30 A(I)=0 : I=I+1 : IF I =N THEN GOTO 30 ELSE END
```

Dans la première phase de la fusion des lignes de notre exemple, nous avons employé le caractère ":" pour concaténer deux lignes (mode ":"). Maintenant, nous inaugurons un autre mode avec le ELSE (que nous appelerons mode "ELSE"). Ce n'est qu'après l'application de ces deux modes que le noyau minimal est parfaitement déterminé.
Mais alors que les règles de concaténation du mode ":" sont relativement simples, celles du mode "ELSE" semblent moins évidentes. Nous nous contenterons donc dans ce qui suit du premier de ces modes qui, s'il ne détermine pas le noyau minimal, le cerne de près.

Voici donc les règles qui permettent de déterminer en mode ":" un noyau dans lequel se trouveront toutes nos lignes vitales. Font partie de ce noyau :

1- la première ligne du programme BASIC
2- toute ligne dont le numéro suit un GOTO, GOSUB ou RESUME
3- toute ligne succédant à un REM
4- toute ligne succédant à un IF
5- toute ligne dont le numéro peut éventuellement suivre un THEN ou un ELSE (équivalent du GOTO).

Toute ligne incluse dans le noyau grâce à un ou plusieurs des critères 1, 2, 3 ou 5 est une ligne vitale. Par contre, celle incluse dans le noyau grâce au critère 4 exclusivement n'est pas forcément une ligne vitale. Ainsi, l'application de la concaténation en mode "ELSE" ne diffère en ses règles que sur le critère 4 (critère + restrictif). Voir pour cela, notre premier exemple.

Voyons maintenant la structure du programme de recherche des lignes vitales. Je ne reviens pas dans cet article sur l'organisation d'un programme BASIC en mémoire centrale, celle-ci étant clairement expliquée dans les numéros précédents de TRACE. Interessons-nous plutôt à la liste du programme proprement dit. D'abord, que fait-il ? Il stocke dans le tableau T(I) les lignes du noyau déterminé par les règles de concaténation en mode ":", et ceci, pour tout programme BASIC. Voici quelques explications.

- Ligne 64000 : L=numéro de ligne, PT=adresse pointeur BASIC, T()=tableau des résultats (lignes vitales).
- Ligne 64001 : 16548 pointe au début de la zone BASIC (programme). F est utilisé comme drapeau (1 si la prochaine ligne à rencontrer va dans le tableau). N=nombre de lignes rangées dans le tableau.
- Ligne 64002 : Est-ce la fin du programme ?
- Ligne 64003 : On pointe l'octet numéro de ligne.
- Ligne 64004 : L prend la valeur du numéro de ligne.
- Ligne 64005 : Est-il déjà prévu de le ranger dans T ?
- Ligne 64006 : Il n'y a plus de raison a priori, que la prochaine ligne soit rangée dans T.
- Ligne 64007 : On pointe au premier octet du "texte" de la ligne BASIC.
- Ligne 64008 : 141 est le token de GOTO, 145=GOSUB, 159=RESUME.
- Ligne 64009 : 143=IF, 147=REM. F=1 : la prochaine ligne rencontrée sera rangée dans le tableau.
- Ligne 64010 : 202=THEN, 149=ELSE.
- Ligne 64011 : On pointe l'octet suivant du "texte" BASIC.
- Ligne 64012 : Est-ce la fin de la ligne ?
- Ligne 64013 : Pointe le 1er octet de la ligne BASIC suivante.
- Lignes 64014 à 64022 : Sous-programme d'insertion de L dans le tableau T des résultats. Pas la peine d'insérer la même valeur plusieurs fois.
- Lignes 64032 à 64040 : 47=code ASCII de "0" et 57=code ASCII de "9".
- Ligne 64037 : 44=token de ",".

Passons maintenant au mode opératoire. Le programme ci-joint étant introduit, frappez "RUN" et, au bout d'un certain temps, le message "READY" apparaît. Tapez alors "?N" pour connaître le nombre de lignes effectivement enregistrées dans le tableau T. Vous devez obtenir 25, ce qui signifie que notre programme de 41 lignes en compte moins de 25 qui soient vitales. Pour obtenir les 25 "élues", il ne reste plus qu'à frapper :

```
FOR I=1 TO N : PRINT T(I); : NEXT
```

Comment utiliser VITAL pour ausculter d'autres programmes que lui-même (car c'est là le but) ? Il faut

le sauver sur cassette ou disquette en ayant soin d' ajouter à la fin de la ligne 64004 :

```
... : IF L>63999 THEN END
```

Le programme à ausculter étant chargé en mémoire (numéro de ligne inférieur à 64000), il est nécessaire de faire un "MERGE" avec VITAL. En version disque, pas de problème avec la commande MERGE à condition que VITAL soit sauvé en ASCII (SAVE "VITAL",A). En version cassette, on peut frapper directement au clavier dans le cas du TRS modèle 1 (et si l'on ne possède pas de routine réalisant cette fonction) :

```
>A=PEEK(16633)+256*PEEK(16634)-2
>POKE 16549,A/256 : POKE 16548,A AND 255
>CLOAD "VITAL"
>POKE 16549,66 : POKE 16548,233
```

A ce stade, les deux programmes doivent figurer en mémoire, l'un à la suite de l'autre. Faire alors :

```
>RUN 64000
(attendre le READY)
>?N : FOR I=1 TO N : PRINT T(I); : NEXT
```

```
64000 DEFINT A-Z : DEFSNG L,P,T : DIM T(40):REM1B
64001 PT=PEEK(16548)+256*PEEK(16549) : F=1 : N=0:REMC3
64002 IF PEEK(PT)+PEEK(PT+1)=0 THEN END:REMA7
64003 PT=PT+2:REME7
64004 L=PEEK(PT)+256*PEEK(PT+1):REMF6
64005 IF F=1 THEN GOSUB 64015:REMCB
64006 F=0:REMB5
64007 PT=PT+2:REME3
64008 P=PEEK(PT) : IF P=141 OR P=145 OR P=159 THEN 64028:REM84
64009 IF P=143 OR P=147 THEN F=1:REM07
64010 IF P=202 OR P=149 THEN 64039:REM4E
64011 PT=PT+1:REME0
64012 IF PEEK(PT)=0 THEN 64013 ELSE 64008:REMF3
64013 PT=PT+1 : GOTO 64002:REM1B
64014 ' Sous-programme d'insertion de L dans T(i) avec O<i<N  F8

64015 IF N=0 THEN 64020:REM4F
64016 G=1:REMA9
64017 FOR I=1 TO N : IF T(I)=L THEN G=0 : I=N:REM86
64018 NEXT I:REM24
64019 IF G=0 THEN 64022:REM50
64020 N=N+1:REM83
64021 T(N)=L:REMDD
64022 RETURN:REM5E
64023 ' Sous-programme filtrant les octets gerant le curseur  EF

64024 PT=PT+1:REMD3
64025 IF PEEK(PT)>0 AND PEEK(PT)<33 THEN 64024:REMCE
64026 RETURN:REM5A
64027 ' Routine de traitement des GOTO, GOSUB et RESUME  EB
64028 GOSUB 64024:REM59
64029 M$="":REM5F
64030 M$=M$+CHR$(PEEK(PT)):REM42
64031 PT=PT+1:REMCC
64032 IF PEEK(PT)>47 AND PEEK(PT)<58 THEN K=K+1 : GOTO 64030:REM
58
64033 L=VAL(M$):REM0D
64034 GOSUB 64015:REM53
64035 ' Ici intervient la possibilite d'un ON.. GOTO ou autre  E
3
64036 GOSUB 64025:REM50
64037 IF PEEK(PT)=44 THEN 64028 ELSE 64012:REMA1
64038 ' Routine de traitement des THEN et ELSE  E0
64039 GOSUB 64024:REM4E
64040 IF PEEK(PT)>47 AND PEEK(PT)<58 THEN 64029 ELSE 64012:REMAC

64050 '------------------------------------------------
64051 ' VITAL - (C) Benoit Gilon et TRACE
64052 ' Configuration : modeles 1, 3 et compatibles
64053 '               16 a 48 K, cassette ou disquette
64054 '------------------------------------------------
```



Benoît Gilon

utilitaire

Et désassemblez maintenant !

Matériel : PC-1500 + CE-150 + CE-151 (ou CE-155)
Programme + données : 4384 octets

Depuis le numéro 5 de TRACE, vous avez appris à connaître l'assembleur de votre machine. Seuls deux outils vous manquent : un désassembleur et un assembleur mnémonique. Je vous propose ici le premier de ces deux programmes :

Il est en BASIC et occupe 1817 octets (et pourtant il est rapide...). Ne possédant pas de module 8K mais seulement un module 4K, je n'ai pas pu intégrer les mnémoniques sous forme de DATA dans le programme mais j'en ai fait un fichier sur cassette (évidemment, le possesseur du CE-155 n'aura pas ce problème).

1ère étape : rentrez le petit programme qui suit, exécutez-le et rentrez toutes les chaînes indiquées sur la fig. 1, puis préparez votre magnéto K7 pour l'enregistrement et appuyez sur ENTER.

```
5000:DIM A$(255)*
     10:WAIT 0
5010:FOR I=0TO 25
     5:PRINT I;"
     ";
5020:INPUT A$(I):
     PRINT :NEXT
     I
5030:WAIT 50:
     PRINT "POSIT
     IONNEZ LA K7
     ":WAIT :
     PRINT "ET AP
     PUYEZ SUR EN
     TER"
5050:PRINT #"FICN
     OR";A$(*)
```

```
0 : 0SBCA,XI
1 : 0SBCA,(X)
2 : 0ADCA,XI
3 : 0ADCA,(X)
4 : 0LDA,XI
5 : 0LDA,(X)
6 : 0CPA,XI
7 : 0CPA,(X)
8 : 0LDXh,A
9 : 0ANDA,(X)
10 : 0LDXI,A
11 : 0ORA,(X)
12 : 0DSBCA,(X)
13 : 0XORA,(X)
14 : 0LD(X),A
15 : 0BITA,(X)
16 : 0SBCA,YI
17 : 0SBCA,(Y)
18 : 0ADCA,YI
19 : 0ADCA,(Y)
20 : 0LDA,YI
21 : 0LDA,(Y)
22 : 0CPA,YI
23 : 0CPA,(Y)
24 : 0LDYh,A
25 : 0ANDA,(Y)
26 : 0LDYI,A
27 : 0ORA,(Y)
28 : 0DSBCA,(Y)
29 : 0YORA,(X)
30 : 0LD(Y),A
31 : 0BITA,(Y)
32 : 0SBCA,UI
33 : 0SBCA,(U)
34 : 0ADCA,UI
35 : 0ADCA,(U)
36 : 0LDA,UI
37 : 0LDA,(U)
38 : 0CPA,UI
39 : 0CPA,(U)
40 : 0LDUh,A
41 : 0ANDA,(U)
42 : 0LDUI,A
43 : 0ORA,(U)
44 : 0DSBCA,(U)
45 : 0UORA,(X)
46 : 0LD(U),A
47 : 0BITA,(U)
48 : 0
49 : 0
50 : 0
51 : 0
52 : 0
53 : 0
54 : 0
55 : 0
56 : 0
57 : 0
58 : 0
59 : 0
60 : 0
61 : 0
62 : 0
63 : 0
64 : 0INCXI
65 : 0LDI(X),A
66 : 0DECXI
67 : 0LDD(X),A
68 : 0INCX
69 : 0LDIA,(X)
70 : 0DECX
71 : 0LDDA,(X)
72 : 1LDXh,n
73 : 1AND(X),n
74 : 1LDXI,n
75 : 1OR(X),n
76 : 1CPXh,n
77 : 1BIT(X),n
78 : 1CPXI,n
79 : 1ADD(X),n
80 : 0INCYI
81 : 0LDI(Y),A
82 : 0DECYI
83 : 0LDD(Y),A
84 : 0INCY
85 : 0LDIA,(Y)
86 : 0DECY
87 : 0LDDA,(Y)
88 : 1LDYh,n
89 : 1AND(Y),n
90 : 1LDYI,n
91 : 1OR(Y),n
92 : 1CPYh,n
93 : 1BIT(Y),n
94 : 1CPYI,n
95 : 1ADD(Y),n
96 : 0INCUI
97 : 0LDI(U),A
98 : 0DECUI
99 : 0LDD(U),A
100 : 0INCU
101 : 0LDIA,(U)
102 : 0DECU
103 : 0LDDA,(U)
104 : 1LDUh,n
105 : 1AND(U),n
106 : 1LDUI,n
107 : 1OR(U),n
108 : 1CPUh,n
109 : 1BIT(U),n
110 : 1CPUI,n
111 : 1ADD(U),n
112 : 0
113 : 0
114 : 0
115 : 0
116 : 0
117 : 0
118 : 0
119 : 0
120 : 0
121 : 0
122 : 0
123 : 0
124 : 0
125 : 0
126 : 0
127 . 0
128 : 0SBCA,Xh
129 : 1JRNC+n
130 : 0ADCA,Xh
131 : 1JRC+n
132 : 0LDA,Xh
133 : 1JRNH+n
134 : 0CPA,Xh
135 : 1JRN+n
136 : 1DJNZ-n
137 : 1JRNZ+n
138 : 0RETI
139 : 1JRZ+n
140 : 0DADCA,(X)
141 : 1JRNV+n
142 : 1JR+n
143 : 1JRV+n
144 : 0SBCA,Yh
145 : 1JRNC-n
146 : 0ADCA,Yh
147 : 1JRC-n
148 : 0LDA,Yh
149 : 1JRNH-n
150 : 0CPA,Yh
151 : 1JRN-n
152 : 0
153 : 1JRNZ-n
154 : 0RET
155 : 1JRZ-n
156 : 0DADCA,(Y)
157 : 1JRNV-n
158 : 1JR-n
159 : 1JRV-n
160 : 0SBCA,Uh
161 : 2SBCA,(W)
162 : 0ADCA,Uh
163 : 2ADCA,(W)
164 : 0LDA,Uh
165 : 2LDA,(W)
166 : 0CPA,Uh
167 : 2CPA,(W)
168 : 0SPV
169 : 2ANDA,(W)
170 : 2LDS,W
171 : 2ORA,(W)
172 : 0DADCA,(U)
173 : 2XORA,(W)
174 : 2LD(W),A
175 : 2BITA,(W)
176 : 0
177 : 1SBCA,n
178 : 0
179 : 1ADCA,n
180 : 0
181 : 1LDA,n
182 : 0
183 : 1CPA,n
184 : 0RPV
185 : 1ANDA,n
186 : 2JMPW
187 : 1ORA,n
188 : 0
189 : 1XORA,n
190 : 2CALLW
191 : 1BITA,n
192 : 0
193 : 1CALLNC,n
194 : 0
195 : 1CALLC,n
196 : 0
197 : 1CALLNH,n
198 : 0
199 : 1CALLH,n
200 : 0
201 : 1CALLNZ,n
202 : 0
203 : 1CALLZ,n
204 : 0
205 : 1CALLn
206 : 0
207 : 1CALLV,n
208 : 2ENTW
209 : 0RRA
210 : 0
211 : 0SRD
212 : 0
213 : 0SRA
214 : 0
215 : 0SLD
216 : 0
217 : 0SLA
218 : 0
219 : 0RLA
220 : 0
221 : 0INCA
222 : 1EVL+n
223 : 0DECA
224 : 0ERR
225 : 0SPU
226 : 0RTS
227 : 0RPU
228 : 0ERR1
229 : 0
230 : 0
231 : 0
232 : 0
233 : 3AND(W),n
234 : 0
235 : 3OR(W),n
236 : 0
237 : 3BIT(W),n
238 : 0
239 : 3ADD(W),n
240 : 0
241 : 0SWP
242 : 0CLS
243 : 0RVF
244 : 0
245 : 0LDI(Y),(X)
246 : 0
247 : 0CPIA,(X)
248 : 0
249 : 0RCF
250 : 0
251 : 0SCF
252 : 0
253 : 0
254 : RST
255 : 0SHFT
```

2ème étape : entrez le programme DESAS.2.0, sauvez-le sur K7 (nul n'est à l'abri d'une perte de programme), exécutez-le ; aux "BIPS", positionnez votre K7 en lecture sur le fichier et appuyez sur ENTER, répondez au point d'interrogation par une adresse de début de désassemblage et laissez tourner...

N.B. : Il n'est pas nécessaire de recharger le fichier à chaque fois, le chargement fait, exécuter le programme par DEF Z.

Architecture du programme :

Lignes 1-55	: 1ère initialisation
Lignes 60-61	: 2ème initialisation
Lignes 70-71	: impression hexa de l'adresse
Lignes 75	: si préfixe &FD ⟶ 2000
Lignes 80-85	: impression hexa des octets
Lignes 87-95	: impression mnémoniques
Ligne 100	: opération avec 1 octet opérande
Lignes 200-203	: opération avec 2 octets opérandes
Lignes 300-307	: opération avec 3 octets opérandes
Lignes 1000-1001	: traduction décimal ⟶ hexa
Lignes 2000-2999	: en cas de préfixe &FD

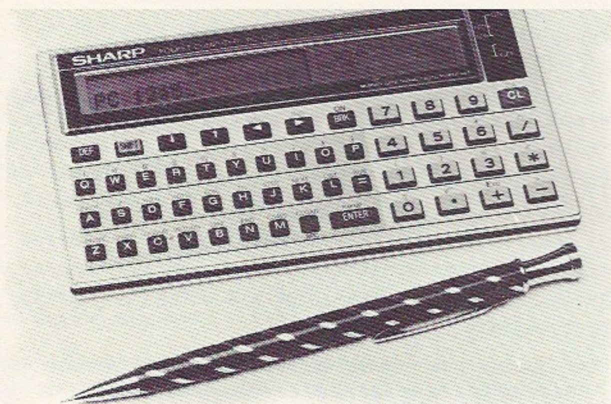
Vous programmiez ? eh bien, désassemblez à présent...

DESAS 2.0

```
1:DIM A$(255)*10
55:BEEP 10:PRINT
  :INPUT #"FICNO
  R";A$(*)
60:"Z"INPUT X:S=8
  5:IF X<0OR X>6
  5535GOTO 60
61:M$="0123456789
  ABCDEF":CSIZE
  1
70:WAIT 0:A=INT (
  X/256):GOSUB 1
  000:LPRINT A$;
71:A=((X/256)-INT
  (X/256))*256:
  GOSUB 1000:
  LPRINT A$;" ";
75:IF PEEK X=&FD
  GOTO 2000
80:K$=A$(PEEK X)
85:GOSUB 800
87:B$=RIGHT$ (K$,
  LEN K$-1)
90:ON VAL K$GOSUB
  100,200,300
95:LCURSOR 16:
  LPRINT B$:X=X+
  VAL K$+1:GOTO
  70
100:B$=LEFT$ (B$,
  LEN B$-1):A=
  PEEK (X+1):
  GOSUB 1000:B$=
  B$+A$:RETURN
200:FOR I=1TO LEN
  B$:IF MID$ (B$
  ,I,1)="W"THEN
  202
201:NEXT I:PRINT "
  ERREUR"
202:Z=I:A=PEEK (X+
  1):GOSUB 1000:
  C$=A$:A=PEEK (
  X+2):GOSUB 100
  0:C$=C$+A$
203:B$=LEFT$ (B$,Z
  -1)+C$+RIGHT$
  (B$,LEN B$-Z):
  RETURN
300:FOR I=1TO LEN
  B$:IF MID$ (B$
  ,I,1)="W"THEN
  302
301:NEXT I:PRINT "
  ERREUR"
302:Z=I:A=PEEK (X+
  1):GOSUB 1000:
  C$=A$:A=PEEK (
  X+2):GOSUB 100
  0:C$=C$+A$
303:B$=LEFT$ (B$,Z
  -1)+C$+RIGHT$
  (B$,LEN B$-Z)
305:A=PEEK (X+3):
  GOSUB 1000
307:B$=LEFT$ (B$,
  LEN B$-1)+A$:
  RETURN
800:FOR I=0TO VAL
  K$:A=PEEK (X+I
  ):GOSUB 1000:
  LPRINT A$;
805:NEXT I:RETURN
1000:P=INT (A/16)
  :A$=MID$ (M$
  ,P+1,1)
1001:Q=((A/16)-P)
  *16:A$=A$+
  MID$ (M$,Q+1
  ,1):RETURN
2000:LPRINT "FD";
  :X=X+1:R=
  PEEK X:IF R=
  &18LET K$="0
  LDX,Y":GOTO
  S
2001:IF R=&28LET
  K$="0LDX,U":
  GOTO S
2002:IF R=&ALET K
  $="0POPX":
  GOTO S
2003:IF R=&1ALET
  K$="0POPY":
  GOTO S
2004:IF R=&2ALET
  K$="0POPX":
  GOTO S
2005:IF R=&40LET
  K$="0INCXh":
  GOTO S
2006:IF R=&50LET
  K$="0INCYh":
  GOTO S
2007:IF R=&60LET
  K$="0INCUh":
  GOTO S
2008:IF R=&42LET
  K$="0DECXh":
  GOTO S
2009:IF R=&52LET
  K$="0DECYh":
  GOTO S
2010:IF R=&62LET
  K$="0DECUh":
  GOTO S
2011:IF R=&48LET
  K$="0LDX,S":
  GOTO S
2012:IF R=&58LET
  K$="0LDX,P":
  GOTO S
2013:IF R=&5ALET
  K$="0LDY,X":
  GOTO S
2014:IF R=&6ALET
  K$="0LDU,X":
  GOTO S
2015:IF R=&4CLET
  K$="0RBF":
  GOTO S
2016:IF R=&4ELET
  K$="0LDS,X":
  GOTO S
2017:IF R=&5ELET
  K$="0JP(X)":
  GOTO S
2018:IF R=&81LET
  K$="0EI":
  GOTO S
2019:IF R=&88LET
  K$="0PUSHX":
  GOTO S
2020:IF R=&8ALET
  K$="0POPA":
  GOTO S
2021:IF R=&98LET
  K$="0PUSHY":
  GOTO S
2022:IF R=&A8LET
  K$="0PUSHU":
  GOTO S
2023:IF R=&AALET
  K$="0LDA,F":
  GOTO S
2024:IF R=&B1LET
  K$="0HALT":
  GOTO S
2025:IF R=&BALET
  K$="0LDA,I":
  GOTO S
2026:IF R=&BELET
  K$="0DI":
  GOTO S
2027:IF R=&C0LET
  K$="0RDP":
  GOTO S
2028:IF R=&C1LET
  K$="0SDP":
  GOTO S
2029:IF R=&C8LET
  K$="0PUSHA":
  GOTO S
2030:IF R=&CALET
  K$="0ADDX,A"
  :GOTO S
2031:IF R=&DALET
  K$="0ADDY,A"
  :GOTO S
2032:IF R=&EALET
  K$="0ADDU,A"
  :GOTO S
2033:IF R=&CELET
  K$="0LDI,A":
  GOTO S
2999:K$=LEFT$ (A$
  (PEEK X),1)+
  "#"+RIGHT$ (
  A$(PEEK X),
  LEN A$(PEEK
  X)-1):GOTO S
```

Daniel Glazman

Un nouveau poquette : le Sharp PC-1245



A première vue, le nouveau POCKET de chez SHARP ressemble beaucoup au PC-1251... En fait, cette machine est son héritière directe : tout d'abord, le microprocesseur est le même (donc compatibilité du langage machine entre PC-1251 et PC-1245)... Quant au BASIC, aux instructions RS-232C près (INSTAT, OUTSTAT, etc...) il est à 99 % identique à celui de son prédécesseur :

Toutes les fonctions BASIC standard sont présentes : INPUT? PRINT, WAIT, PAUSE, USING, IF-THEN, NOT et la possibilité, avec le signe &, de traiter des nombres héxadécimaux.

Les opérateurs alphanumériques et numériques sont aussi au complet ; on regrettera seulement que les fonctions trigonométrique hyperboliques ne soient pas présentes.

Du côté des entrées-sorties, le PC-1245 est doté d'un affichage de 16 caractères et d'un clavier "QWERTY" de 52 touches. Comme sur le PC-1251, il existe sur le côté droit de l'appareil un règlage du contraste de l'affichage.

En mémoire, le PC-1245 est pourvu de 24K de ROM et de 2,2K de RAM utilisateur.

Mais la grande nouveauté de ce pocket est (pour SHARP) la disparition du MODE RESERVE (assignation des touches) au profit de touches préprogrammées : en effet 18 touches du clavier sont assignées (sans possibilité de changement) aux 18 fonctions BASIC principales. Ex : F → GOTO, M → CSAVE, etc...

Du côté des extensions, le PC-1245 ne peut pour l'instant être relié qu'au CE-125 (utilisé également sur le PC-1251).

Il s'agit d'un combiné imprimante/magnétophone à microcassettes. L'imprimante est thermique (on pourrait le regretter car les listings sur papier thermique ne sont pas impérissables...) et commandée par les ordres LLIST et LPRINT (intégrés au PC-1245). Le magnétophone utilise des microcassettes standard SONY et est commandé par les ordres CSAVE, CLOAD, CLOAD?, CHAIN, PRINT#, INPUT# et MERGE (intégrés au PC-1245). Le combiné PC-1245+CE-125 tient parfaitement dans une pochette ou une mallette (dimensions du combiné : 135 x 70 x 9,5 mm).

Daniel Glazman

Un autre clavier pour votre poquette

La nouvelle interface CE-153 pour le PC-1500 ressemble énormément à une tablette digitaliseuse. Il ne s'agit pourtant que d'un clavier sensitif de 140 touches. Ses dimensions réduites (170 x 230 mm), permettent à l'ensemble PC-1500 de conserver son caractère "poquette".

Avec la tablette, SHARP fournit une cassette contenant le logiciel d'exploitation, un câble PC-1500 - CE-153, un manuel d'utilisation et 5 caches en plastique.

La cassette contient 3 programmes : le programme d'exploitation qui doit être chargé en mémoire avant toute utilisation de la tablette, et deux programmes d'application (Sales management et Word master).

Le câble de raccordement est assez impressionnant : il est revêtu d'une véritable "cotte de mailles" qui le rend très solide, tout en restant très souple. Le seul point à regretter est sa longueur (40 cm) qui n'est pas suffisante.

Le manuel d'utilisation (malheureusement en anglais) n'est pas un exemple de clarté (programmes décrits avant utilisation de la tablette).

En ce qui concerne l'utilisation, elle ne se fait pas à l'aide de "véritables" instructions BASIC, mais par un CALL... 3 types d'appel sont possibles : GET (attente d'un caractère), INKEY (saisie d'un caractère au vol), et simulation de clavier. Le manuel donne ainsi trois exemples de programmes : Constellations, Dessinateur, Inventaire.

Les caches en plastique s'utilisent avec les programmes d'application.

Un élément déterminant est le prix de cette interface : environ 1500 FTTC. C'est très cher, surtout quand on sait que le CE-153 ne contient que 3 circuits...

Cette tablette est donc utile à toute personne désirant se munir d'un second clavier ou pour certains jeux notament (échecs, othello) où elle peut servir de damier.

Daniel Glazman

Lorsqu'une fonction s'annule...

Quel élève de terminale C ne s'est jamais retrouvé impuissant à résoudre une équation comme $\frac{1}{x} = e^x$??? Après avoir tout essayé, l'élève prend sa calculatrice scientifique et cherche une solution approximative. C'est ce que propose ici : d'après la méthode de Newton, si x est une valeur approché de x_o (solution de l'équation à résoudre f(x)=o), alors $(x - \frac{f(x)}{f'(x)})$ est une meilleure approximation de x_o. En bouclant le calcul sur lui-même, on arrive assez vite à une valeur satisfaisante de x_o.
J'utilise dans mon programme un "auto-programmateur" qui me permet d'introduire ma fonction pendant l'exécution du programme (celle-ci est à introduire en fonction de X). Le programme s'exécute par DEF F.

Ensuite, le programme vous demande une valeur x de départ pour le calcul, puis une sensibilité ε telle que si x est une bonne approximation de x_o alors $|x-x_o| \leqslant \varepsilon$.

Si le programme n'arrive pas à se stabiliser autour d'une valeur, c'est que l'équation n'admet pas de solutions.

En cas d'"ERROR 38", 2 possibilités :

1) pas de solution à l'équation
2) sensibilité ε trop fine ($10^{-9} \leqslant \varepsilon \leqslant 10^{-1}$): réduire la sensibilité en augmentant ε

Et maintenant, la solution de l'équation $\frac{1}{x} = e^x$ est 0,5671432914...

Pour changer de valeur de départ, ne faites pas DEF F mais DEFG : vous gardez la même fonction.

```
(C) Daniel GLAZMAN
(C) TRACE
   Function"s zero

500:"f"FX=*********
    **************
    **************
    **************
    **************
    ******
520:"F"RESTORE "f"
    :I=(PEEK 30910
    -128)*256+PEEK
    30911+6:INPUT
    "FONCTION = ?"
    ;J$:CALL &F957
530:J=31676:FOR I=
    ITO I-10+PEEK
    (I-7):IF PEEK
    J<>13POKE I,
    PEEK J:J=J+1:
    NEXT I
535:POKE I,58,241,
    153,13
540:"G"INPUT "X ?"
    ;X,"SENSIBILIT
    E ?";S
545:WAIT 0
550:GOSUB "f":Y=FX
    :X=X+S:GOSUB "
    f":Z=(FX-Y)/S
560:X=X-Y/Z:PRINT
    X:GOTO 550
```

Daniel Glazman

Bogues, trucs, erreurs (2e cuvée)

I - Entrez le programme suivant :

```
 5:"Z"WAIT 0:
   GCURSOR 0:FOR
   I=0TO 155:
   GPRINT I;:NEXT
   I
10:GOTO 10
```

puis en mode RUN, tapez un texte (par exemple J'ADORE TRACE), puis 2 fois ◀ et DEF Z.

Appuyez sur BREAK et ◀
Hé oui ! le curseur continue à fonctionner mais uniquement sur le caractère où il se trouvait avant le DEF Z...

II - La commande RUN efface les variables de tableaux et la pile, alors comment éviter celà sans passer par une touche utilisateur (DEF Z) ?
Eh bien, tout simplement en tapant GOTO ENTER .... ça marche...

Daniel Glazman

Jumpman, le saute-fantômes

Toujours atteint d'une maladie rare appelée la "pacmanie aigüe", je vous présente ici une nouvelle version du jeu de saute-mouton...

Le jeu est simple : au début les jumpmen sont à gauche, les fantômes à droite, le but est d'inverser cette situation. Pour ce faire, lire la suite de l'article (si, si !).

Le terrain est représenté par dix cases numérotées de gauche à droite et de 0 à 9 (une barre horizontale représente une case vide).

Les jumpmen ne peuvent se déplacer que de gauche à droite et les fantômes de droite à gauche ; voici (pour le jumpman) les mouvements autorisés (et les seuls !) :



Lorsque les jumpmen occupent les cases 7 à 9 et les fantômes les cases 0 à 2, le jeu est fini ; répondez 10 au point d'interrogation vous demandant un numéro de case.

L'architecture du programme à présent :

Lignes 1 à 3	: c'est toujours nous...
Lignes 5 à 15	: initialisation et création du terrain
Ligne 16	: impression du terrain
Ligne 17	: entrée du mouvement
Lignes 18 à 202	: analyse de la validité du coup ; s'il est valable, l'exécuter

```
   1:REM (C) Daniel
          GLAZMAN
   2:REM (C) TRACE
   3:REM PARIS 83
   5:WAIT 0:CLEAR :
     RESTORE
  10:DATA "7E3B7F7F
     3B7E0000000000
  11:DATA "40404040
     40404040400000
  12:DATA "1C3E7F7F
     772200000000
  13:DIM A$(2)*21:
     READ A$(2),A$(
     0),A$(1)
  14:DIM A(9):FOR I
     =1TO 3:A(I)=1:
     NEXT I
  15:FOR I=6TO 8:A(
     I)=2:NEXT I
  16:CLS :FOR I=0TO
     9:GPRINT A$(A(
     I));:NEXT I
  17:CURSOR 18:
     INPUT A:IF A=1
     0GOTO 1000
  18:W=W+1:IF A(A)=
     0GOTO 16
  19:GOTO 100*A(A)
 100:IF A(A+1)=0LET
     A(A+1)=1:A(A)=
     0:GOTO 16
 101:IF A(A+2)=0AND
     A(A+1)<>0LET A
     (A)=0:A(A+2)=1
     :GOTO 16
 102:GOTO 16
 200:IF A(A-1)=0LET
     A(A-1)=2:A(A)=
     0:GOTO 16
 201:IF A(A-2)=0AND
     A(A-1)<>0LET A
     (A)=0:A(A-2)=2
     :GOTO 16
 202:GOTO 16
1000:FOR I=0TO 2:
      IF A(I)=0:
      GOTO 2000
1001:NEXT I
1002:FOR I=7TO 9:
      IF A(I)=0
      GOTO 2000
1003:NEXT I
1010:CLS :WAIT :
      PRINT "En";W
      ;" coups..."
1011:GOTO 5
2000:CLS :WAIT 75
      :PRINT "Vous
       moquez-vous
       de moi ?
2001:WAIT 0:GOTO
      16
```

Lignes 1000 à 2000 : en cas de fin de partie

Et voilà... En sachant qu'il suffit de préciser à votre poquette la case où se trouve le jumpman (ou le fantôme) pour que celui-ci se déplace automatiquement, amusez-vous bien...

Daniel Glazman

Voici les 2 découvertes que j'ai faites :

1) Gain de 197 octets :

Il suffit de faire :

- POKE &7866,0
- POKE &7868,0

MEM affichera alors (sans extension) :
2047
NEW ne changera rien mais NEW 0 redonnera MEM=1850.

2) Il est bien connu que l'on peut protéger ou déprotéger un programme par LOCK et UNLOCK ce qui n'empêche le listage que si le mode est différent de PRO, et l'exécution que si il est différent de RUN. Faire LOCK en mode "Réserve" empêche donc les 2 possibilités mais laisse encore à l'utilisateur la faculté de détruire les touches préprogrammées...
Il existe une autre façon d'empêcher le listage et l'exécution grâce à un POKE bien placé :

POKE &7867,63 (ou tout nombre positif et inférieur à 63) ;
LIST et RUN ne donneront plus rien ;
MEM retrouvera des valeurs très grandes (si l'on fait POKE &7867,0 on aura MEM=18409), mais ces indications sont fantaisistes.
En effet, l'utilisateur sera dans l'impossibilité de programmer (tout essai donnera ERROR 13 bien que l'on ait MEM > 0)
Pour rappeler le programme il suffira de faire POKE &7867,64 (même le pointeur est gardé intact).
Exemple :

Mode "PRO" NEW 0
10 PRINT "O.K., CA MARCHE"
RUN donnera ce que l'on attend.

Si l'on repasse en mode "PRO" et que l'on appuie sur ↓ ou ↑ on verra que le signe - est placé après le deuxième guillemet.
Faire POKE &7867,63 puis ↓ ou ↑ : rien n'apparaît ! De plus, MEM=2281 (!!!) et RUN ne donne rien. Faire POKE &7867,64 puis RUN et "O.K., CA MARCHE" apparaît sur l'afficheur.
Cette astuce plus un LOCK sur RUN, laisse seulement à l'utilisateur la possibilité d'utiliser le PC.2 en calculette...

Paul Courbis

La ville dans la poche

Si l'on admet qu'une rue qui serait fermée aux deux bouts ne mènerait pas bien loin, on discerne le parti à tirer d'un ordinateur sachant ce qui se trouve à chaque bout de cette rue : il va nous dire où elle mène, et ce qui y aboutit.

Le programme "PLAN" est donc surtout composé d'une très longue liste de DATA ; ma version concerne 133 rues du centre de La Rochelle, et chaque rue doit y figurer sous la forme suivante : "Nom", N, N1, N2, où N est le numéro d'ordre de la rue "Nom" dans la liste, N1, celui de la rue qui y mène et N2 celui de la rue où elle aboutit, le choix entre mener et aboutir pouvant être dictée par les sens uniques, si la carte utilisée les indique, ou arbitraire.

Ceux qui aiment leur ville et croient la connaître auront peut-être quelques surprises en réalisant son plan informatisé : ils s'apercevront peut-être qu'ils ignorent le nom d'endroits où ils passent chaque jour.

Le programme proprement dit pourrait très bien commencer en ligne 15 ; pour ma part, j'ai affecté GOTO 10 @ à la touche A, et tout le contenu de la ligne 10 à la touche Z, en mode Réserve. Si c'est aussi votre cas, amusez-vous, en mode pro, à taper un numéro de ligne, puis Shift A : le caractère @ a disparu. Etonnant, non ?

Mais revenons à la ligne 15 : on y dimensionne un tableau de 2 variables alphanumériques, et 3 de 2 variables numériques, correspondant à ce que j'indiquais plus haut. Le programme, faisant semblant de croire qu'une ville n'est faite que de rues, demande à l'utilisateur le nom de celle qu'il recherche et le range en B$(Ø).

Puis il effectue une première lecture des données. J'ai dit qu'il y avait, dans ma version, 133 rues, et la boucle est de 1→135. La première et la dernière série de données servent dans le cas d'impasses, de rues hors des limites choisies, de fautes d'orthographe. Lorsque le programme à repéré dans les données la rue demandée, il stocke son nom et les 3 données numériques qui l'accompagnent dans l'élément 1 de chaque tableau. Il affiche le nom trouvé, puis, grâce à I=135, passe à la suite...

J'ai dit que le programme, par l'INPUT en ligne 20, feignait de croire à la seule existence des rues : en réalité, dans les data, tout ce qui est square, place, esplanade, ou autre bizarrerie de l'urbanisme, est repéré par "*␣", d'où la double formule de repérage en ligne 35.

Cela étant fait, commence une nouvelle lecture de toutes les données : les tests concernent les données numériques. La rue d'où part celle recherchée est suivie du signe →, celle où elle aboutit précède le même signe, celles trouvées en chemin sont entre thèses.

Codez d'abord l'inconnu Ø,Ø,Ø. N'oubliez pas qu'il faudra une <FIN>, codée N,Ø,Ø. Prenez le plan de la ville à soumettre à ce traitement, et délimitez le secteur qui vous intéresse. Souvent, les rues du centre sont repérées dans l'index par un signe distinctif. Préparez bien votre codage en 2 temps : d'abord, les numéros d'ordre, puis les entrées/sorties. Gare aux carrefours : c'est sur la précision de votre code que se joue la validité du programme.

Enfin, si vos données sont correctes, et que vous ayez travaillé sur le plan de Helsinki ou de Sofia, où vous vous rendez bientôt, dites-vous que vous éviterez des conversations laborieuses dans une langue ma foi pas évidente...

```
  5:REM ***  <PLAN>  ***
    *** J.-F. CHAPIN ***
 10:PRINT = PRINT :
    INPUT "IMPRIMANTE ";
    U$: IF ASC U$=79
    PRINT = LPRINT
 15:CLEAR : WAIT : DIM B
    $(1)*22,C(1),D(1),E(
    1)
 20:INPUT "RUE ";B$(0)
 25:RESTORE
 30:FOR I=1 TO 135: READ
    B$(1),C(1),D(1),E(1)
 35:IF B$(0)=B$(1) OR B$
    (0)= MID$ (B$(1),3,
    LEN B$(1)) BEEP 1:
    PRINT B$(1);" :":I=1
    35
 40:NEXT I
 45:RESTORE
 50:FOR I=1 TO 135: READ
    B$(0),C(0),D(0),E(0)
 55:IF D(1)=C(0) BEEP 1:
    PRINT B$(0);" ->":
    GOTO 70
 60:IF E(1)=C(0) BEEP 1:
    PRINT "-> ";B$(0):
    GOTO 70
 65:IF D(0)=C(1) OR E(0)
    =C(1) BEEP 1: PRINT
    "(";B$(0);")"
 70:NEXT I: WAIT 1:
    PRINT ""
 75:INPUT "ENCORE ? ";K$
    : IF ASC K$=79 GOTO
    15
 80:END
800:DATA "<JE NE SAIS OU
    >",0,0,0,"ABLOIS",1,
    124,112,"ABREUVOIR",
    2,98,48
801:DATA "ADMYRAULT",3,3
    1,57,"ALBERT 1ER",4,
    128,93,"ALIENOR",5,1
    3,0
802:DATA "AMELOT",6,110,
```

```
73,"HOTEL DE VILLE",
7,25,73,"ARCERE",8,4
,65
803:DATA "ARSENAL",9,72,
130,"AUFREDY",10,105
,30,"AUGUSTINS",11,3
0,115
804:DATA "BARBOT",12,83,
0,"LECLERC",13,128,0
,"BAZOGES",14,30,115
,"* BARENTIN",15,45
805:DATA 129,"BEARNOISE"
,16,96,132,"BEURRE",
17,73,115,"BLETTERIE
",18,92,52
806:DATA "BONNES FEMMES"
,19,33,87,"BONPLAND"
,20,122,59,"BRAVE RO
NDEAU",21,122,37
807:DATA "BUFFETERIE",22
,59,49,"BUJAUD",23,1
15,73,"* CACAUD",24,
83,37
808:DATA "* CAILLE",25,1
20,7,"CANARDS",26,12
4,112,"CARMES",27,12
7,40
809:DATA "CHAINE",28,40,
108,"* ST JEAN D^ACR
E",29,28,100,"CHAUDR
IER",30,84,128
810:DATA "CHEF DE VILLE"
,31,91,98,"CLOCHE",3
2,98,127,"CLOUTIERS"
,33,69,49
811:DATA "COLLEGE",34,4,
49,"COMMANDERIE",35,
121,84,"COMTESSE",36
,43,104
812:DATA "CORDELIERS",37
,24,0,"CORDERIES",38
,122,103,"CORDOUAN",
39,83,87
813:DATA "* WILSON",40,1
5,28,"DELAYANT",41,4
,99,"DEUX MOULINS",4
2,100,118
814:DATA "DUC",43,50,46,
"DUPATY",44,84,7,"*
DUPERRE",45,124,15
815:DATA "* DURAND",46,1
24,102,"ECOLES",47,1
28,99,"ESCALE",48,31
,10
816:DATA "EVESCOT",49,34
,22,"FABRIQUE",50,10
4,124,"FAGOTS",51,79
,98
817:DATA "FERTE",52,7,72
,"FLEURIAU",53,30,11
5,"FONDERIES",54,72,
37
818:DATA "FORME",55,122,
59,"FOURCHE",56,112,
116,"FROMENTIN",57,4
8,84
819:DATA "GALLIENI",58,6
,122,"GAMBETTA",59,6
9,103,"GARGOULLEAU",
60,30,69
820:DATA "GENTILSHOMMES"
,61,25,73,"GLACIERE"
,62,93,12,"GRILLE",6
3,44,52
821:DATA "GROSSE HORLOGE
",64,91,15,"JAILLOT"
,65,34,83,"LADAUGE",
66,116,36
822:DATA "LA NOUE",67,98
,13,"LA ROCHELLE",68
,73,133,"* MARCHE",6
9,115,87
823:DATA "MARIETTES",70,
115,73,"MASSIOU",71,
4,12,"* MAUBEC",72,1
10,114
824:DATA "MERCIERS",73,1
8,69,"MERMOZ",74,86,
0,"MERVAULT",75,34,7
7
825:DATA "METZ",76,13,0,
"MINAGE",77,19,4,"MO
LIERE",78,0,76
826:DATA "MONNAIE",79,10
8,0,"MOUREILLES",80,
114,18,"MURAILLES",8
1,83,71
827:DATA "GUILBAUD",82,7
6,74,"ORBIGNY",83,4,
24,"PALAIS",84,30,91
828:DATA "PARADIS",85,11
6,43,"PARC",86,78,74
,"PAS DU MINAGE",87,
39,69
829:DATA "SUEDE",88,13,0
,"PAULEON",89,34,39,
"PERNELLE",90,128,10
830:DATA "* PETITS BANCS
",91,84,64,"PORT",92
,45,114,"PORTE DAUPH
INE",93,4,0
831:DATA "PORTE NEUVE",9
4,13,98,"PRECHE",95,
9,110,"RAMBAUD",96,4
,47
832:DATA "* RASTEAU",97,
3,48,"REAUMUR",98,51
,67,"* REIMS",99,41,
47
833:DATA "REMPART",100,4
2,94,"REMPART VOILIE
RS",101,37,0,"REMPAR
T ST CLAUDE",102,46
834:DATA 124,"PORTE ROYA
LE",103,59,0,"ST CLA
UDE",104,50,46
835:DATA "ST COME",105,9
8,128,"ST DOMINIQUE"
,106,49,83,"ST FRANC
OIS",107,110,103
```

Exemple d'exécution :

comment aller de la rue d'Ablois
à la rue du Temple.
Le choix qui rallonge (St Nicolas)
se trouve automatiquement corrigé.

```
ABLOIS :
-> ST NICOLAS
* VALIN ->

ST NICOLAS :
(ABLOIS)
(CANARDS)
* DURAND ->
-> FABRIQUE
(FOURCHE)
(SARDINERIE)

* DURAND :
(DUC)
-> REMPART ST CLAUDE
(ST CLAUDE)
(ST NICOLAS)
(SARDINERIE)
(SOLETTE)
* VALIN ->

* VALIN :
(ABLOIS)
(CANARDS)
-> * DUPERRE
(* DURAND)
FABRIQUE ->
(REMPART ST CLAUDE)

* DUPERRE :
-> * BARENTIN
(PORT)
* VALIN ->

* BARENTIN :
(* WILSON)
* DUPERRE ->
(GROSSE HORLOGE)
-> VIELJEUX

GROSSE HORLOGE :
-> * BARENTIN
* PETITS BANCS ->

* PETITS BANCS :
(CHEF DE VILLE)
-> GROSSE HORLOGE
PALAIS ->
(TEMPLE)
```

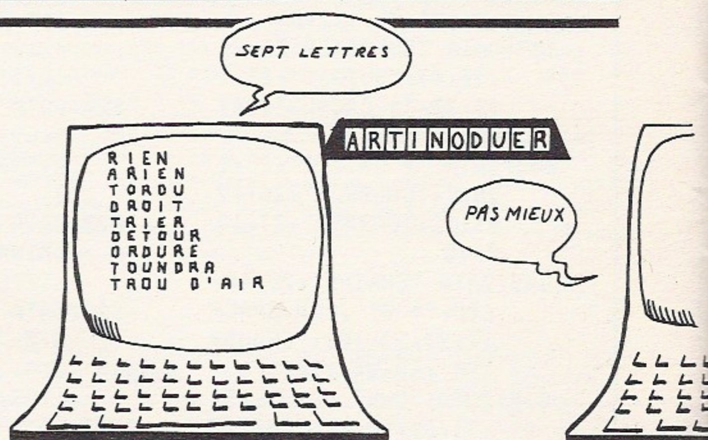
```
836:DATA "ST JEAN",108,4
0,79,"ST LEONARD",10
9,2,31,"ST LOUIS",11
0,72,37
837:DATA "ST MICHEL",111
,52,73,"ST NICOLAS",
112,46,50,"STS PERES
",113,127,98
```

```
838:DATA "ST SAUVEUR",11
     4,72,25,"ST YON",115
     ,69,7,"SARDINERIE",1
     16,46,112
839:DATA "SOLETTE",117,4
     6,116,"SUR LES MURS"
     ,118,28,42,"TALLEMAN
     D",119,10,2
840:DATA "TEMPLE",120,91
     ,25,"TEMPLIERS",121,
     120,44,"THIERS",122,
     69,37
841:DATA "TROIS FUSEAUX"
     ,123,55,6,"* VALIN",
     124,50,45,"VAUCLERC"
     ,125,96,128
842:DATA "VENETTE",126,1
     0,48,"VERDIERE",127,
     108,31,"* VERDUN",12
     8,30,47
843:DATA "VIELJEUX",129,
     15,0,"VILLENEUVE",13
     0,72,37,"VOILIERS",1
     31,62,37
844:DATA "STES CLAIRES",
     132,99,4,"DAMES",133
     ,6,122,"<FIN>",134,0
     ,0
```

J.F. Chapin

jeu

Un mot peut en cacher un autre



Il s'agit de trouver un mot comme au pendu, avec une différence : l'ordinateur n'a pas de réserve de mot et il est nécessaire d'être deux, l'un donne le mot l'autre le cherche. Si une lettre est représentée elle apparaîtra automatiquement partout où il se doit. Les lettres du début et de la fin ne sont pas données au départ.

Le programme sur PC-1211 se divise en 2 parties :

Mode DEF SHFT A : l'ordinateur demande tout d'abord le nombre de lettres du mot (maximum 12 lettres) puis le mot, lettre par lettre.

SHFT B : l'ordinateur indique le nombre de lettres qui composent le mot puis demande le niveau de difficulté : 0 = autant d'essais que de lettres, 4 = 4 essais en plus du nombre de lettres (on peut choisir d'autres valeurs). Puis une fois le niveau défini il faut donner une première lettre.
A la suite de quoi, l'ordinateur affiche une suite de "?" (autant qu'il y a de lettres) et remplace au fur et à mesure les "?" par les lettres correctes.

Notez qu'après l'affichage du mot en formation il est nécessaire d'appuyer sur (ENTER) avant de passer à l'essai suivant. Cet "inconvénient" peut être supprimé sur un 1211 en remplaçant l'ordre PRINT par une boucle et l'ordre PAUSE. Sur le 1210 ce n'est pas possible (manque de place).

Une fois les essais terminés et si le mot n'est pas trouvé, l'ordinateur indique quel était ce mot.

Comme il se présente, le programme nécessite les mémoires A - Z + A(27) A(28) A(29) et occupe 374 pas (ce qui en fait en tout 398).

```
(C) ANGEL ETIENNE
(C) TRACE
  Mot cache...

  1:"A"CLEAR :
    INPUT "NB LETT
    RES(MAX 12)=?"
    ;Z
  2:FOR Y=1TO Z:A$
    (Y)="?":NEXT Y
    :FOR Y=13TO 12
    +Z:INPUT A$(Y)
    :NEXT Y:END
  3:"B"PAUSE "MOT
    EN ";Z;" LETT.
    ":INPUT "NIV(F
    =4,D=0)";A(27)
  5:FOR Y=1TO Z+A(
    27):INPUT A$(2
    9)
  6:FOR A(28)=13TO
    (Z+12):IF A$(2
    9)=A$(A(28))
    LET A$(A(28)-1
    2)=A$(29)
  7:NEXT A(28)
  8:PRINT A$;B$;C$
    ;D$;E$;F$;G$;H
    $;I$;J$;K$;L$:
    NEXT Y
  9:FOR Y=1TO Z:IF
    A$(Y)="?"PAUSE
    "PERDU":GOTO 1
    5
 10:NEXT Y
 12:BEEP 5:PRINT "
    GAGNE":END
 15:PAUSE "LE MOT
    ETAIT":BEEP 3:
    PRINT M$;N$;O$
    ;P$;Q$;R$;S$;T
    $;U$;V$;W$;X$:
    END
```

Etienne Angel

COPY

(suite)

Le Modèle 1 possède plusieurs Systèmes d'Exploitation des Disquettes (SED, ou DOS en anglais). Ces différents systèmes ont beaucoup de points communs, particulièrement ceux de la "lignée" TRSDOS-NEWDOS-NEWDOS80-LDOS, qui s'inspirent tous, à quelques détails près, du TRSDOS de Tandy (version initiale 2.1). Nous publierons régulièrement sous forme de fiches pratiques, les commandes de ces différents DOS, classées alphabétiquement. Ces fiches ne prétendent pas remplacer les manuels originaux qu'il est impératif de posséder (il n'était pas envisageable de publier sous cette forme l'équivalent de plusieurs centaines de pages ...).

Leur but est de fournir à l'utilisateur une information rapide d'accès (classement alphabétique), pratique (exemples d'utilisation), comparative (différents DOS) et dans une langue qu'il est censé connaître (français)...

Les DOS les plus répandus ont été retenus pour constituer ces fiches. Ce sont le TRSDOS version 2.3 de Tandy, le NEWDOS80 version 2 d'Apparat et le LDOS version 5.0 de Lobo Drives International. En règle générale, nous nous baserons sur les commandes du TRSDOS, en signalant les différences ou les apports rencontrés dans les autres systèmes d'exploitation.

Conventions d'écriture

Dans la description des commandes, les mots en caractères majuscules désignent les commandes proprement dites, ainsi que les mots-clés associés. Les mots en caractères minuscules ne font pas partie des commandes mais en représentent la forme. Au moment de l'exécution, il faudra leur substituer le ou les mots appropriés. Parmi ceux-ci, nous trouverons par exemple : "nomfich", qui désigne un nom de fichier, "motpasse" qui signifie qu'un mot de passe (password) est attendu à cet endroit. Dans tous les cas, un coup d'oeil sur les exemples donnés devrait lever toute ambiguité.
Enfin, il faut signaler que le NEWDOS80 emploie indifféremment la virgule ou l'espace comme séparateur dans les commandes.

Nous terminons dans ce numéro la description des différents formats de la commande COPY sous NEWDOS 80 V2.0 par l'étude du format 6.

FORMAT 6

```
COPY,[:]dn1[,TO],[:]dn2[=tc2],mm/jj/aa,CBF[,Y][,N]
     [,USR][,/ext][,UPD][,ILF=specfich3][,XLF=specfich4]
     [,CFWO][,NDMW][,FMT][,NFMT][,SPDN=dn3]
     [,DPDN=dn4][,SPW=mot-de-passe1][,ODPW=mot-de-passe2]
     [,NDPW=mot-de-passe3][,DDND][,ODN=nom1][,KDN]
     [,KDD][,NDN=nom2][,SN=nom3][,USD][,UBB]
     [,DDSL=ln1][,DDGA=gc1]
```

Ce format réalise une "copie" de fichiers multiples et se distingue du format 5 par l'utilisation du paramètre CBF (COPY by FILE). Le choix des fichiers à recopier est fait par l'utilisation conjuguée des options : USR, EXT, UPD, ILF, XLF, CFWO. Les options du format 5 (voir TRACE n° 6) à l'exception de BDU sont réutilisées dans le format 6.

Si NFMT est spécifié alors aucun des paramètres suivants ne peuvent être utilisés dans le format 6 : Y, N, KDN, KDD, ND,N, BDU, USD, NDPW, DDSL, DDGA, TC2.
Si NFMT n'est pas
nation est formattée comme si la commande était FORMAT avec l'implantation initiale de BOOT/SYS et DIR/SYS, puis la mise au "Répertoire" (DIR/SYS) de tous les fichiers à recopier.

Les paramètres spécifiques du format 6 sont :

CBF (Copy by file)	Cette option indique que la copie doit s'effectuer par fichier plutôt que séquentiellement par ordre des secteurs de la disquette.
USR (User)	Seuls les fichiers "utilisateurs" sont recopiés (Fichiers figurant au répertoire visible) ; les fichiers invisibles et systèmes (/SYS) sont exclus.
/EXT	Seuls les fichiers portant l'extension /EXT sont recopiés.
UPD (Updated files)	Seuls les fichiers ayant le drapeau de "MISE A JOUR" positionné sont recopiés. Ce drapeau est établi par la routine d'écriture standard SED pour indiquer qu'au moins un secteur a été écrit ou réécrit depuis que le drapeau de mise à jour a été remis à zéro. Ce drapeau est remis à Ø par une demande spécifique établie par PROT et ATTRIB. Cette RAZ n'a pas lieu lors de l'ordre COPY.
ILF = SpecFich3 (Include List File)	SpecFich3 spécifie un fichier contenant une liste de fichiers à recopier ; si un fichier n'est pas dans la liste il n'est pas copié. A l'intérieur de la liste chaque fichier à copier doit être spécifié par son nom complet (avec extension) suivi par un caractère de fin de ligne (EOL) = ØDH. Si une spécification commence par un point virgule cette ligne est considérée comme un commentaire et non prise en compte. Chaque spécification excepté un commentaire est limitée à un maximum de 13 caractères incluant EOL = ØDH. ILF est mutuellement exclusif avec XLF.
XLF = SpecFich 4 (Exclusion	SpecFich 4 spécifie un fichier contenant une liste de fichiers à ne pas prendre en compte dans la recopie, les

List File)	caractéristiques de ce fichier sont identiques à celles de ILF. XLF est mutuellement exclusif avec ILF.
CFWO (Check File with Operator)	Pour qualifier les fichiers à recopier le SED demande à l'utilisateur un fichier à la fois, par un message écran affichant le nom de fichier ; répondre Y si la copie doit être prise en compte N dans le cas contraire ; répondre R pour redémarrer l'ensemble de la séquence d'interrogation ou Q pour quitter la commande. Aucun fichier n'est copié tant que l'interrogation n'est pas terminée.
ODPW = mot de passe 2 (Old Destination Diskette Password)	Si NFMT est spécifié, si les mots de passe sont validés et si l'option système AR = N, alors la copie nécessite une disquette destination avec un mot de passe correspondant. Si le mot de passe 2 ne correspond pas avec celui de cette disquette destination alors la copie est interrompue.
DDSL = ln1 (Destination Diskette Directory Starting Lump)	Si DDSL est specifié le formattage fait démarrer le "repertoire" au premier secteur du lump ln1 autrement le n° de de lump de démarrage par défaut est utilisé (voir commande PDRIVE). DDSL est mutuellement exclusif avec NFMT.
DDGA = gc1 (Destination Diskette Directory Granule Allocation)	Le formattage alloue gc1 ($2 \leqslant gc1 \leqslant 6$) granules au repertoire si DDGA est spécifié. Autrement c'est la spécification par défaut de PDRIVE qui est utilisé. DDGA est mutuellement exclusif avec NFMT.

Remarques : Si à la suite d'une copie en format 6 la disquette destination a insuffisamment d'espace pour contenir un fichier, le message "DISKETTE FULL = nom/ext) et le marqueur de fin de fichier (EOF) du fichier destination est mis à zéro.

Une copie avec une seule unité de disquette ne peut être effectuée sous Format 6 (ou Format 5) par "appel SED" (DOS-CALL) sous BASIC par exemple, car la copie sous DOS-CALL se limite à la zone mémoire en-dessous de 7ØØOH, ceci nécessitant alors un nombre d'échanges de disquettes trop important.

Exemples :

```
COPY,0,1,,CBF,CFWO,NDMW
```

Cette commande recopie les fichiers de la disquette 0 sur la disquette 1 pour lesquels l'opérateur répond "Y" (yes) à la question posée (paramètre CFWO). NDMW sous-entend que les disquettes sont montées dans les unités 0 et 1 et le message "monter la disquette source ... destination ..." n'est pas affiché. Remarquez que la date est sautée par les les deux virgules.

```
COPY,0,1,,CBF,USR,DFDN=0
```

Ici, tous les fichiers visibles (paramètre USR) sont copiés de la disquette 0 vers la disquette 1, cette dernière possédant les caractéristiques définies par le PDRIVE de l'unité 0.

Index des mots-clés

La liste des mots-clés figurant ci-dessous est destinée à être exploitée par le programme de gestion des articles décrit dans le numéro 1 de TRACE (pages 58 à 64). Le format est le suivant:

Nom abrégé de l'article, numéro de page : 1 à 4 mots-clés.

En prenant l'habitude d'entrer cette liste à chaque numéro de TRACE, vous constituerez ainsi une petite base de données vous permettant de trouver plus efficacement et plus rapidement l'article que vous cherchez. Bien entendu, cette liste est donnée à titre indicatif, et vous pourrez toujours en modifier ou en compléter le contenu, afin de l'adapter à vos besoins ; mais avant toute chose, vous devez vous imposer une certaine normalisation des mots-clés (en principe, ils sont limités à 12 caractères) PROGRAMME par exemple, indiquera toujours qu'un listage de programme est donné dans l'article. Pour cela aussi, faites-nous part de vos idées!

KIT ET DOUBLE, 19 : ESSAI, DISQUETTES, MATERIEL, KIT
MONSTRES, 20 : PROGRAMME, BASIC, JEU, MATHEMATIQUES
PARLER-VOUS, 24 : PROGRAMME, BASIC, ENSEIGNEMENT
RUBANS, 28 : BRICOLAGE, MATERIEL, IMPRIMANTE
GARDEZ TRACE, 29 : PROGRAMME, MOD1, MOD3, OUTILS
MINI-ZAP, 32 : PROGRAMME, OUTILS, DOS
REFLEXION, 35 : PROGRAMME, BASIC, JEU
PLUS VITE, 38 : PROGRAMME, OUTILS, CASSETTE
AMELIOREZ, 43 : PROGRAMME, OUTILS, CASSETTE
AMORCE, 45 : PROGRAMME, OUTILS, CASSETTE
SESAME, 46 : PROGRAMME, OUTILS, DOS
OUVREZ, 47 : BRICOLAGE, MATERIEL, INTERFACE
EXTENSIONS, 48 : OUTILS, EXTENSION, BASIC, CASSETTE
TRI, 51 : PROGRAMME, OUTILS, TRI
LIGNES VITALES, 55 : PROGRAMME, OUTILS, PROGRAMMATION
DESASSEMBLEZ, 57 : POQUETTE, PROGRAMME, ASSEMBLEUR, PC1500
PC1245, 59 : POQUETTE, PC1245, ESSAI
CLAVIER, 59 : POQUETTE, CE153, ESSAI
FONCTION, 60 : POQUETTE, PROGRAMME MATHEMATIQUES, ENSEIGNEMENT
BOGUES, 60 : POQUETTE, PROGRAMME
JUMPMAN, 61 : POQUETTE, PROGRAMME, JEU
VILLE, 62 : POQUETTE, PROGRAMME, PRATIQUE
UN MOT, 64 : POQUETTE, PROGRAMME, JEU
DOS DE A A Z, 65 : FICHE DOS

Imprimerie CEPI, 75010 Paris - Imprimé en France - Directeur de la publication : Jean-Pierre Nizard
Dépôt légal : Novembre 1983

PENTA SONIC

PENTA 8 34, rue de Turin, 75008 PARIS - Tél. 293.41.33
- Métro : Liège, St-Lazare, Place Clichy - Télex 614789

PENTA 13 10 bd Arago, 75013 PARIS - Tél. 336.26.05
- Métro : Gobelins (service correspondance et magasin)

PENTA 16 5 rue Maurice Bourdet, 75016 PARIS 524.23.16
(pont de Grenelle) - Metro Charles Michels - Bus 70/72 : Maison de l'ORTF

HORAIRES : du lundi au samedi
de 9 heures à 19.30 sans interruption *Sauf PENTA 8 qui ferme à 19 heures.



MICROFAZER

Buffer d'imprimante de 16 jusqu'à 128 K



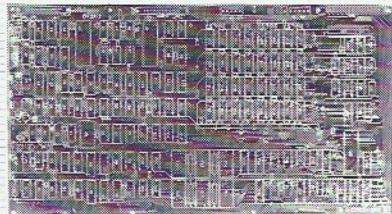
Cet interface se branche directement sur la ligne // ou série de votre imprimante et permet la buffarisation de vos données. Cela veut dire que quelle que soit la vitesse du printer (un nodem, plotter), après quelques secondes votre ordinateur redeviendra disponible, les données à transmettre n'étant plus dans votre RAM mais dans la RAM du Microfazer.

Monté, testé 16 K //→// .... **2310 F**

128 K //→// **3970 F**

Existe en version série → série

SPECIAL PROF 80



Le C.I. et les plans **647 F**

Prof 80 est un circuit imprimé double face, trous métallisés avec vernis épargne et sérigraphie. Il est disponible au prix de 647 F TTC et une fois monté, vous donne accès à toute la bibliothèque de programmes du TRS 80®.
Tous les composants du PROF 80 sont disponibles chez PENTA 8, 13 ou 16.
A titre indicatif le BASIC 12 K est vendu 357 F.

CARACTERISTIQUES :
- CPU Z80 4 MHz.
- 64 k RAM (dont 16 k Shadow pour CP/m).
- 12 K Basic LNW 80®.
- Interface cassette standard TRS 80®.
- Interface parallèle type EPSON.
- Interface série type RS232C et 20 mA.
- Clavier AZERTY ou QWERTY.
- Sortie vidéo et UHF (modulateur en option).

- Interface floppy 5'' 40 ou 96 TPI. 1 à 4 lecteurs.
- Compatible TRS DOS®, L DOS®, NEW DOS®, OS 80®.

OPTIONS :
- Carte graphique 8 couleurs matrice 256 x 512 sortie Péritel 48 K RAM contrôleur 9366 Efcis. 456 F (le CI seul).
- Carte CP/M 229 F (CI seul).

- Doubleur de densité. Permet de travailler en 5'' en double densité. Monté, testé .... 1397 F

SERVICE CORRESPONDANCE : Commandez par téléphone

Demandez Catherine au 336.26.05 avant 16 h votre commande partira le jour même*

** en fonction des stocks disponibles*

SYNTHETISEUR DE VOIX POUR TRS 80® OU PROF 80

COMPLET MONTE TESTE avec disquette .... **495 F**

Ce synthétiseur travaille sur le principe des phonèmes.
Vous tapez sur votre clavier
— BØNJØUR JE SUI LE PRØF KATR VIN...
— Run... et vous entendez une voix synthétique qui vous dit «Bonjour je suis le PROF 80».

FLOPPY DISQUES



5''

SF-SD. Avec anneau de renforcement	22,50
DF-DD 96 TPI	33,00
SF-DD 10 sect.	43,00
SF-SD 16 sect.	43,00
DF-DD 16 sect.	44,00

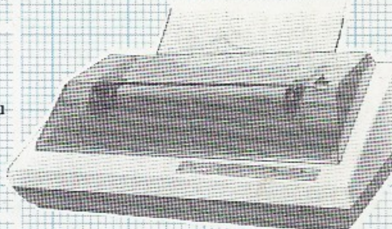
8''

SF-DD	44,00
DF-DD	54,00

CARTE GRAPHIQUE COULEUR

Donnez la dimension couleur à votre TRS 80® modèle I ou modèle III, vidéo génie ou PROF 80.
Caractéristiques : matrice 256 x 512, 8 couleurs, branchement direct sur le bus.

Montée testée avec disquette **2458 F**



SEIKOSHA GP 100

Imprimante graphique compacte - Interface parallèle en standard - 80 car./ligne - 50 car./sec. - Impression en simple ou double largeur - Papier normal - Entraînement par tracteurs ajustables - Interfaces TRS 80®, PET, RS 232, APPLE II disponibles.

GP100. Papier 10''.
Promotion .... **2480 F**

DES DOUBLEURS DE DENSITE POUR UN TRS 80®

LE DOUBLEUR SEUL **1397 F TTC**

Cet interface se monte en quelques minutes et vous permet de doubler la capacité de vos floppys. D'origine PERCOM, ce doubleur est livré avec la disquette «OS 80 D» et manuels. Une fois installé le doubleur vous procure une capacité disque de 180 K par lecteur et permet le transfert de tous vos programmes simple densité.

DRIVE FLOPPY NOUVEAU HALF SIZE



AVERTISSEMENT :
Les lecteurs de disque nécessitent des réglages d'azimutage très précis et, en conséquence, supportent très mal les transports. C'est pourquoi les lecteurs achetés chez Pentasonic seront testés devant vous au moment de votre achat et ce gratuitement.
De plus pendant 45 jours ils pourront être révisés et réglés sur place (Penta 16) également gratuitement.

Lecteurs simple face double densité hauteur normale ou demi-hauteur .... 2195 F
Double face double densité .... 2995 F
Double face double densité 96 TPI Half Size... 3795 F

Les nouveaux Half Size sont chez Pentasonic et vendus au même prix que les normaux.
Tavernier, Prof 80, TRS 80®, etc.
* Il est possible de monter le 96 TPI sur un TRS 80® sur un Tavernier et sur un PROF 80.

CARTE EXTENSIONS FLOPPY POUR TRS 80®

Pour modèle I : MDX II. Le CI et les plans .... **735 F**
Pour modèle III : MDX III. Le CI et les plans ...... **725 F**
MDX VI. Monté et testé .... **1497 F**
MDX « mécanique » avec alimentation . **1382 F**



& OTHER MYSTERIES

DISPONIBLES CHEZ PENTA

Captain 80	TRSDOS 2.3
TRS 80 Disk	Custom Apple
Microsoft BASIC	Cyberchess System
Custom TRS 80	Professional #1
BASIC Faster & Better	Professional #2
Machine Language Disk 1/10	Professional #3



DOS PLUS

DOS PLUS est un des dos les plus performants existant pour TRS 80 modèle I et III.
Démonstration chez PENTA 16.

FANTASTIQUE **1119 F**